

Universidad de las Ciencias Informáticas



Facultad 1

**Título:** Sistema de gestión de la evaluación del desempeño de los trabajadores en la facultad 1.

**Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.**

**Autores:** Lilianny Marrero Travieso  
Reynaldo Lázaro Batista Vázquez

**Tutor:** Ing. Damián Cervantes Rodón

Ciudad de la Habana

Junio de 2009

***Profundizar en el conocimiento científico es una de las mejores vías para lograr plenitud y libertad.***

***Pilar Álvarez Pellicero***

## AGRADECIMIENTOS

*A mi familia por todo su apoyo durante estos cinco años, su cariño y por estar siempre ahí para mí...*

*A la Revolución por esta gran oportunidad...*

*Al 1108, mis compañeros desde el primer día...*

*Un agradecimiento especial a todos esos que me han apoyado y comprendido durante todos estos años...mis amigos.*

**Lily**

*A los que todo me lo han dado, mis padres.*

*A David, mi amigo y maestro.*

*A mi flaca lindisísima que nunca se rindió.*

*A Dayron por su disposición y consejos*

*A René, Heber, Frank, Céspedes, y el flaco médico por querer lo mejor para un amigo.*

*A todos los que me regalaron su granito de arena.*

*A todos los que en Holguín están esperando por un ingeniero.*

**Rey**

*A nuestro tutor por todo su apoyo...*

**Los dos**

## DEDICATORIA

*A mis padres por ser las personas más importantes de mi vida, mi guía, mi motivación, por ser mi ejemplo siempre. Gracias por todo su amor, comprensión, sacrificio y dedicación. Los quiero mucho.*

*A mi abuelita que es una persona muy especial en mi vida...*

**Lily**

*A mi gran familia que siempre confió en mí.*

*A todos mis brothers.*

**Rey**

## RESUMEN

La evaluación del desempeño de los profesores en la Universidad de las Ciencias Informáticas es un proceso de gran importancia, ya que de su correcto funcionamiento depende la estimulación que le es otorgada a los trabajadores de la misma. Dada la gran cantidad de personas a evaluar que existe actualmente, se maneja de una gran cantidad de información. Hoy en día este proceso se realiza de forma manual, por lo que el proceso de evaluación resulta muy difícil de realizar. En la facultad el proceso se realiza de igual forma, resultando muy complicado para las personas que se encargan de realizar la evaluación. De esta manera surgió la necesidad de crear un sistema informático que gestionara toda la información necesaria para realizar la evaluación del desempeño de los trabajadores. Con este trabajo se tiene como objetivo diseñar e implementar todo el proceso de evaluación, de forma tal que le permita a los evaluadores tener toda la información necesaria sobre cada persona y todas las tareas que ha realizado en cierto período de tiempo a la hora de realizar la evaluación. Para el diseño del sistema se utilizaron herramientas que respetan la política de migración a software libre que tienen tanto el país como la universidad, se utilizó la herramienta de modelado Visual Paradigm 6.1, guiados por la metodología RUP y UML como lenguaje de modelado. También se tuvieron en cuenta el lenguaje de programación PHP5 y el marco de trabajo Symfony para darle soporte a este lenguaje.

**Palabras Claves:** desempeño, evaluación, trabajadores.

## ÍNDICE DE CONTENIDO

INTRODUCCIÓN .....	1
CAPÍTULO 1.....	6
1.1.    Introducción.....	6
1.2.    Estado del arte de los procesos de evaluación del desempeño .....	6
1.2.1.    Sistemas de gestión de la evaluación del desempeño en el mundo.....	6
1.2.2.    Sistemas de gestión de la evaluación del desempeño en Cuba .....	9
1.3.    Tendencias actuales.....	12
1.3.1.    Software libre.....	12
1.4.    Metodologías para el desarrollo de software .....	12
1.4.1.    Metodologías pesadas o altamente predictivas .....	13
1.4.2.    Metodologías ligeras.....	14
1.5.    Lenguaje y herramientas para el modelado de software .....	15
1.6.    Tecnologías para la implementación de software .....	16
1.6.1.    Lenguajes utilizados en aplicaciones web.....	16
1.6.2.    Marcos de trabajo utilizados para el desarrollo web.....	20
1.6.3.    Sistemas gestores de base de datos .....	22
1.7.    Fundamentación metodológica .....	23
1.8.    Conclusiones.....	25
CAPÍTULO 2.....	26
2.1.    Introducción.....	26
2.2.    Principales reglas del negocio .....	26
2.3.    Actores del negocio .....	26
2.4.    Trabajadores del negocio.....	27
2.5.    Diagrama de casos de uso del negocio.....	27
2.6.    Descripción textual de los casos de uso del negocio .....	27

---

2.7.	Requerimientos funcionales .....	29
2.8.	Requerimientos no funcionales .....	42
2.9.	Actores del sistema .....	42
2.10.	Diagrama de casos de uso del sistema .....	43
2.11.	Descripción abreviada de los casos de uso del sistema .....	44
2.12.	Estimación de esfuerzo .....	48
2.13.	Conclusiones .....	56
CAPÍTULO 3.....		57
1.1.	Introducción.....	57
1.2.	Diagramas de clases de análisis.....	57
1.3.	Aplicación de algunos patrones GRASP en Symfony .....	66
1.4.	Descripción de las clases del diseño .....	67
1.5.	Diagramas de clases del diseño .....	71
1.6.	Diseño de la base de datos .....	78
1.7.	Modelo de despliegue .....	82
1.8.	Tratamiento de errores.....	83
1.9.	Conclusiones.....	84
CAPÍTULO 4.....		85
4.1.	Introducción.....	85
4.2.	Modelo de Implementación.....	85
4.2.1.	Diagrama de componentes .....	85
4.3.	Marco de trabajo de pruebas.....	92
4.4.	Descripción de los casos de prueba de integración .....	95
4.5.	Conclusiones.....	99
CONCLUSIONES .....		100
RECOMENDACIONES .....		101
BIBLIOGRAFÍA REFERENCIADA .....		102
BIBLIOGRAFÍA CONSULTADA .....		103

GLOSARIO DE TÉRMINOS .....	106
ANEXOS.....	108



---

## ÍNDICE DE TABLAS

Tabla 2.1. Descripción de actores del negocio.....	26
Tabla 2.2. Descripción de trabajadores del negocio.....	27
Tabla 2.3: Descripción textual del caso de uso “Entregar Plan de Trabajo”.....	28
Tabla 2.4. Descripción textual del caso de uso “Modificar Plan de Trabajo”.....	28
Tabla 2.5. Descripción textual del caso de uso “Realizar Evaluación”.....	29
Tabla 2.6. Actores del sistema.....	42
Tabla 2.7. Resumen del caso de uso “Gestionar evaluación”.....	44
Tabla 2.8. Resumen del caso de uso “Gestionar área”.....	44
Tabla 2.9. Resumen del caso de uso “Gestionar cargo”.....	44
Tabla 2.10. Resumen del caso de uso “Gestionar usuario”.....	45
Tabla 2.11. Resumen del caso de uso “Gestionar usuario local”.....	45
Tabla 2.12. Resumen del caso de uso “Gestionar tipo de tarea”.....	45
Tabla 2.13. Resumen del caso de uso “Gestionar parámetro”.....	45
Tabla 2.14. Resumen del caso de uso “Gestionar tarea evaluada”.....	46
Tabla 2.15. Resumen del caso de uso “Gestionar certificado”.....	46
Tabla 2.16. Resumen del caso de uso “Gestionar tipo de evaluación”.....	46
Tabla 2.17. Resumen del caso de uso “Gestionar tarea”.....	46
Tabla 2.18. Resumen del caso de uso “Gestionar tareas especiales”.....	47
Tabla 2.19. Resumen del caso de uso “Asignar Tareas”.....	47
Tabla 2.20. Resumen del caso de uso “Asignar funcionalidades”.....	47
Tabla 2.21. Resumen del caso de uso “Asignar cargos”.....	47
Tabla 2.22. Resumen del caso de uso “Asignar áreas”.....	48
Tabla 2.23. Resumen del caso de uso “Asignar cargo evaluado”.....	48
Tabla 2.24. Resumen del caso de uso “Autenticar usuario”.....	48
Tabla 2.25. Factor de peso de los casos de uso sin ajustar.....	49
Tabla 2.26. Factor de peso de los casos de uso sin ajustar.....	50
Tabla 2.27. Factor de complejidad técnica.....	52
Tabla 2.28. Factor de ambiente.....	53
Tabla 2.29. Distribución del Esfuerzo.....	54

---

Tabla 2.30. Descripción de la clase myUser. ....	67
Tabla 2.31. Descripción de la clase actions del módulo de autenticación.....	68
Tabla 2.32. Descripción de la clase actions del módulo de información.....	68
Tabla 2.33. Descripción de la clase actions del módulo de errores. ....	69
Tabla 2.34. Descripción de las clases actions de los módulos construidos para gestionar los nomencladores de la base de datos. ....	70
Tabla 2.35. Descripción de las clases actions de los módulos que gestionan las asignaciones en el sistema. ....	71
Tabla 4.1. Pruebas de la clase TbRevaluadorEvaluadoPeer.....	92
Tabla 4.2. Pruebas de la clase TbRtareaEvaluadaPeer. ....	92
Tabla 4.3. Pruebas de la clase soap.class.php. ....	94
Tabla 4.4. Pruebas de la clase TbRevaluadorEvaluadoCargoAreaPeer. ....	94
Tabla 4.5. Descripción del caso de prueba Gestionar evaluación. Sección “Adicionar evaluación”. ....	95
Tabla 4.6. Descripción del caso de prueba Gestionar evaluación. Sección “Modificar evaluación”. ....	95
Tabla 4.7. Descripción del caso de prueba Gestionar área. Sección “Adicionar área”. ....	95
Tabla 4.8. Descripción del caso de prueba Gestionar área. Sección “Modificar área”. ....	96
Tabla 4.9. Descripción del caso de prueba Gestionar cargo. Sección “Adicionar cargo”. ....	96
Tabla 4.10. Descripción del caso de prueba Gestionar cargo. Sección “Modificar cargo”. ....	97
Tabla 4.11. Descripción del caso de prueba Gestionar usuario. Sección “Adicionar usuario”. ....	97
Tabla 4.12. Descripción del caso de prueba Gestionar usuario. Sección “Modificar usuario”. ....	97
Tabla 4.13. Descripción del caso de prueba Gestionar tarea evaluada. Sección “Adicionar tarea evaluada”. .....	98
Tabla 4.14. Descripción del caso de prueba Gestionar tarea evaluada. Sección “Modificar tarea evaluada”. .....	98

## ÍNDICE DE FIGURAS

Figura 1. Diagrama de casos de uso del negocio.....	27
Figura 2. Diagrama de casos de uso del sistema.....	43
Figura 3. Diagrama de clases de análisis del caso de uso “Gestionar evaluación”.....	58
Figura 4. Diagrama de clases de análisis del caso de uso “Gestionar área”.....	58
Figura 5. Diagrama de clases del análisis del caso de uso “Gestionar cargo”.....	59
Figura 6. Diagrama de clases del análisis del caso de uso “Gestionar usuario”.....	59
Figura 7. Diagrama de clases del análisis del caso de uso “Gestionar usuario local”.....	60
Figura 8. Diagrama de clases del análisis del caso de uso “Gestionar tipo de tarea”.....	60
Figura 9. Diagrama de clases del análisis del caso de uso “Gestionar parámetro”.....	61
Figura 10. Diagrama de clases del análisis del caso de uso “Gestionar tarea evaluada”.....	61
Figura 11. Diagrama de clases del análisis del caso de uso “Gestionar certificado”.....	62
Figura 12. Diagrama de clases del análisis del caso de uso “Gestionar tipo de evaluación”.....	62
Figura 13. Diagrama de clases del análisis del caso de uso “Gestionar tarea”.....	63
Figura 14. Diagrama de clases del análisis del caso de uso “Gestionar tarea especial”.....	63
Figura 15. Diagrama de clases del análisis del caso de uso “Asignar tareas”.....	64
Figura 16. Diagrama de clases del análisis del caso de uso “Asignar funcionalidades”.....	64
Figura 17. Diagrama de clases del análisis del caso de uso “Asignar cargo”.....	64
Figura 18. Diagrama de clases del análisis del caso de uso “Asignar cargo evaluado”.....	65
Figura 19. Diagrama de clases del análisis del caso de uso “Asignar áreas”.....	65
Figura 20. Diagrama de clases del análisis del caso de uso “Autenticar usuario”.....	65
Figura 21. Diagrama de clases del diseño del caso de uso “Gestionar Evaluación”.....	72
Figura 22. Diagrama de clases del diseño del caso de uso “Gestionar Área”.....	73
Figura 23. Diagrama de clases del diseño del caso de uso “Gestionar Cargo”.....	74
Figura 24. Diagrama de clases del diseño del caso de uso “Gestionar Usuario”.....	75
Figura 25. Diagrama de clases del diseño del caso de uso “Gestionar Tarea Evaluada”.....	76

Figura 26. Diagrama de clases del diseño del caso de uso “Gestionar Tarea” .....	77
Figura 27. Modelo lógico de datos. ....	79
Figura 28. Modelo físico de datos. ....	81
Figura 29. Modelo de despliegue. ....	83
Figura 30. Diagrama de componentes del caso de uso “Gestionar usuario”. ....	86
Figura 31. Diagrama de componentes del caso de uso “Gestionar cargo”. ....	87
Figura 32. Diagrama de componentes del módulo “Gestionar área”. ....	88
Figura 33. Diagrama de componentes del módulo “Gestionar evaluación” .....	89
Figura 34. Diagrama de componentes del módulo “Gestionar tarea”. ....	90
Figura 35. Diagrama de componentes del módulo “Gestionar tarea evaluada”. ....	91
Figura 36. Fragmento de código para las clases TbRevaluadorEvaluadoPeer y TbRtareaEvaluadaPeer. ....	93
Figura 37. Fragmento de código para la clase soap.class.php. ....	94
Figura 38. Fragmento de código para la clase TbRevaluadorEvaluadoCargoAreaPeer. ....	94

## INTRODUCCIÓN

Las tecnologías de la información y las comunicaciones constituyen una importante diferencia entre una civilización desarrollada y otra en vías de desarrollo. Éstas poseen la característica de ayudar a comunicarnos ya que, a efectos prácticos, en lo que a captación y transmisión de información se refiere, desaparecen el tiempo y la distancia geográfica.

La actividad tecnológica influye en el progreso social y económico, siendo la tendencia actual a la innovación y desarrollo de los medios tecnológicos o informáticos, una necesidad para almacenar, procesar y difundir todo tipo de información: visual, digital o de otro tipo con diferentes finalidades. Por tanto se puede afirmar que el uso de instrumentos tecnológicos es una prioridad en la comunicación de hoy en día, y que las tecnologías de la comunicación son una importante forma de dar soporte a diferentes procesos dentro de la sociedad, pudiendo transformar la realidad de muchas personas en el mundo.

Cuba ha estado inmersa en el profundo y novedoso proceso de transformaciones educacionales y sociales como son los programas de la Batalla de Ideas, a partir de la cual se iniciaron y se emprenden nuevos programas destinados a elevar el nivel cultural de la población y su calidad de vida.

De esta forma en el año 2002 fue creada la Universidad de las Ciencias Informáticas (UCI), un nuevo modelo de universidad, por primera vez puesto en práctica en Cuba. El objetivo principal de este proyecto es formar ingenieros especializados en la informática, que se encuentren actualizados tecnológicamente y también crear una nueva generación de profesionales comprometidos con la Revolución, capaces de responder incondicionalmente ante los extraordinarios retos que les serán impuestos durante y al finalizar su carrera.

Para introducir el tema a tratar se debe comenzar explicando en qué consiste la evaluación del desempeño, la misma no es más que un proceso sistemático y periódico de comparación entre el desempeño de una persona en su trabajo y las pautas definidas por la organización en que brinde servicios. Es un medio de apreciación del desempeño del individuo en el cargo que éste desempeña y su potencial de desarrollo.

Debido al gran aumento que tiene la UCI en cantidad de trabajadores, se hace más difícil poder tener el control de todos los datos que se necesitan conocer para realizar la evaluación del desempeño de los mismos, además de que el evaluado no cuenta con la posibilidad de chequear periódicamente esta evaluación, es por esta razón que se necesita un sistema informático dedicado a la gestión de la evaluación del desempeño de los trabajadores en la facultad.

Por lo que se plantea el siguiente **problema a resolver**: ¿Cómo lograr una adecuada gestión de los procesos de la evaluación del desempeño de los trabajadores en la facultad 1?

Para darle solución al problema definido se plantean las siguientes **preguntas científicas**:

- ¿Cómo se gestionan las evaluaciones del desempeño de los trabajadores en la facultad 1?
- ¿Cómo se podría plantear alguna vía confiable para el manejo y control de esta información en la facultad 1?
- ¿De qué forma se implementaría algún proceso de seguimiento de las evaluaciones del desempeño de los trabajadores en la facultad 1?

El **objeto de estudio** de la investigación es el proceso de gestión de la evaluación del desempeño.

El **campo de acción** queda enmarcado específicamente en el proceso de gestión y almacenamiento de la evaluación del desempeño de los trabajadores.

Definiendo como **objetivo general** de la investigación: Realizar el diseño e implementación de una propuesta de sistema para la automatización de los procesos de evaluación del desempeño de los trabajadores. Y como **objetivos específicos**:

- Realizar un estudio de los diferentes sistemas de gestión de evaluación del desempeño.

- Realizar el modelado de los procesos del negocio.
- Obtener los requerimientos de la propuesta de solución.
- Diseñar e implementar los procesos de gestión de evaluación del desempeño.
- Diseñar y ejecutar las pruebas unitarias necesarias que aseguren la calidad de la propuesta de solución.

Para cumplir con los objetivos trazados se desarrollaron las siguientes **tareas**:

- Estudiar el estado del arte de los procesos de gestión de evaluación del desempeño.
- Elaborar la fundamentación teórica de todo el proceso.
- Diseñar los procesos involucrados al módulo.
- Implementar las funcionalidades necesarias del sistema
- Diseñar y ejecutar las pruebas que aseguren la calidad de la propuesta de solución.

Como **hipótesis** se tiene que si se implementa un sistema de gestión de la evaluación del desempeño entonces se podrá mejorar la gestión de esta información en la facultad.

De la hipótesis anterior se derivan las siguientes variables:

**Variable Independiente:** Implementación del Sistema de Gestión de la Evaluación del desempeño de los trabajadores.

**Variable Dependiente:** Mejora de la gestión de la información.

**Idea a defender:** El análisis, diseño e implementación de un sistema informático para gestionar la evaluación del desempeño de los trabajadores, permitirá automatizar el desarrollo de este proceso y facilitará el trabajo de los evaluadores.

Como **métodos** de investigación científica se utilizaron:

**Métodos Teóricos:**

- **Histórico-Lógico:** Posibilitó el análisis histórico del proceso de gestión de evaluación del desempeño.
- **Método Analítico-Sintético:** La utilización de este método posibilitó analizar el proceso de evaluación del desempeño del trabajador.

- **Método Modelación:** Se han creado modelos que permiten representar las características y relaciones fundamentales del fenómeno.

### **Métodos Empíricos:**

- **Método de observación:** Para la recolección de la información este es un método muy factible y sumamente utilizado. Después de un exhausto desarrollo de los métodos teóricos quedan muy bien definidos los aspectos significativos para llevar a cabo el registro visual del fenómeno en cuestión. Es decir se ha previsto un esquema de cuáles son los factores y acontecimientos importantes y cuáles son menos o no significativos a tener en cuenta.
- **Método Entrevista:** Se realizaron entrevistas con el fin de definir el problema a resolver específicamente y establecer el objeto de estudio. Las entrevistas fueron tanto individuales como colectivas.

**Resultados Esperados:** Un sistema que logre brindar las funcionalidades al cliente-usuario y que se adapte a las necesidades de la facultad. Además de la documentación del sistema de desarrollo que se genera.

### **Estructura de la investigación:**

**Capítulo #1:** Fundamentación teórica: Contiene la fundamentación teórica del tema. Se realiza un estudio del estado del arte. Se ejemplifica el proceso utilizando referencias de procesos similares. Se mencionan las tendencias, técnicas, tecnologías, metodologías y softwares usados en la actualidad o en las que se apoya para la solución del problema que se enfrenta.

**Capítulo #2:** Características del Sistema: Contiene las características fundamentales del sistema. Se realiza la ingeniería del software, definiéndose el modelo de negocio, los actores, trabajadores, entidades que intervienen en el mismo, las actividades a automatizar y se describen los casos de uso del negocio. Se define además la propuesta de sistema analizando los requerimientos funcionales y no funcionales. Identificando para el mismo los actores y casos de uso y la descripción de estos. Finalmente se calcula la estimación del esfuerzo que requiere la realización del desarrollo de la propuesta.



**Capítulo #3:** Análisis y Diseño del Sistema: Contiene los modelos de Análisis y Diseño del sistema respectivamente. Se definen las clases de análisis del sistema. Concluyendo con el análisis breve del tratamiento de errores y la concepción general de la ayuda.

**Capítulo #4:** Implementación y prueba: Contiene los diagramas de despliegue y de componente, así como una breve descripción de estos conceptos. De igual forma se reflejan las pruebas de caja blanca y caja negra que certifican la calidad del sistema propuesto.

## CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

### 1.1. Introducción

En este capítulo se propone realizar una investigación referente a los distintos sistemas de evaluación del desempeño existentes en Cuba y en el mundo. Además se va a realizar un análisis detallado acerca de las tendencias, herramientas y tecnologías actuales, que pueden ajustarse a la problemática que se plantea.

### 1.2. Estado del arte de los procesos de evaluación del desempeño

#### 1.2.1. Sistemas de gestión de la evaluación del desempeño en el mundo

##### **EVAL 360**

Sistema automatizado de evaluación y gestión de desempeño para empresas "EVAL 360".

Este sistema consiste en una plataforma que permite evaluar desempeño en línea con los objetivos y metas trazadas de los diferentes colaboradores (principio básico de toda evaluación desempeño), además calcular el cumplimiento de estas mismas a lo largo del año, mostrando gráficamente sus avances, ingresar el plan para el año siguiente y nuevamente medirlo.

Entre sus utilidades están que se puede medir los resultados iniciales para cumplir con el plan estratégico que la empresa se ha propuesto.

Permite trabajar con diversos instrumentos de medición asociados a cada perfil de colaboradores, y con informes automatizados y disponibles en tiempo real.

Este software ya ha sido implementado en empresas como Falabella, Cencosud Chile, entre otros. (1)

##### **Sistema informático para la evaluación laboral**

Desarrollado por estudiantes del Instituto Politécnico Nacional (IPN), es un sistema informático que facilita la evaluación de desempeño laboral, el cual será instalado en oficinas de la Procuraduría General de la República para mejorar el funcionamiento del personal de la dependencia.

El sistema busca un mayor control del desempeño profesional y el incremento de la transparencia del proceso.

Instalado en el servidor de Internet de esa dependencia, y estará disponible en todas las entidades del territorio nacional mediante el sistema Intranet.

El programa informático se estructuró en cinco módulos: el primero permite introducir los datos de la persona que se va a evaluar.

Existe otro módulo al cual sólo tiene acceso el evaluador, quien aplica una serie de cuestionarios y califica el desempeño del candidato conforme al conocimiento, evidencias, y aplicación correcta de las técnicas y metodologías a operar.

En los módulos restantes que conforman el sistema, es posible cotejar la adecuada aplicación de exámenes, revisar paso a paso la evaluación y alimentar la base de datos con las normas, instrumentos y procesos confidenciales que requiera la institución.

El sistema se diseñó para las necesidades de evaluación de la PGR y los usuarios del sistema sólo tendrán acceso a la información de su competencia. Hay datos restringidos a los que sólo tiene acceso el personal autorizado.

El último módulo del sistema incluye la información confidencial de la dependencia.(2)

### **QTRAINING**

#### Detección de Necesidades de Capacitación

Este software presenta en forma de reportes los resultados de la evaluación de desempeño. La evaluación de desempeño que a su vez es la mejor manera de evaluar la competencia del personal, le va a decir quienes necesitan capacitación y en cuales temas.

Logra administrar de forma eficiente su presupuesto de capacitación, porque en lugar enviar a un grupo de trabajadores a un curso, ahora solamente enviará a los trabajadores que realmente lo necesitan y en vez de contratar un curso general, Usted solicitará al instructor los temas específicos que necesita su empresa.

La administración y programación de los cursos, así como los participantes se registran en QTraining, que luego emitirá las listas de asistencias y reportes.

Simplemente aplica otra evaluación de desempeño con los mismos indicadores establecidos y la diferencia entre los resultados antes y después del curso le dirán si efectivamente el curso impactó en una mejora del desempeño del trabajador. Y así tiene que ser, o para qué se entrena al personal sino es para que tenga un mejor desempeño en la empresa.

Es registrada toda la información de las evaluaciones de desempeño y los cursos de capacitación. Esta información se puede consultar y filtrar en cualquier momento a través de un potente generador de consultas avanzadas que le permite construir prácticamente cualquier consulta sobre la información registrada en la base de datos.

### Requerimientos mínimos

- Sistema operativo: MS Windows XP con Service Pack 2, MS Windows Server 2003 con Service Pack 1 o MS Windows 2000 con Service Pack 4
- Procesador: Pentium o compatible a 600 MHz
- RAM: 192 MB
- Monitor: SVGA
- Mouse: MS Mouse
- Unidad de CD-ROM (3)

### Evaluación de Desempeño para el Desarrollo Docente

Características de la herramienta:

- Posibilidad de seleccionar una modalidad de evaluación desde los modelos siguientes:
  - Alternativa 1: Evaluación de las competencias funcionales y conductuales (modelo integrado).
  - Alternativa 2: Evaluación de las competencias funcionales (modelo funcional).
- Flexibilidad: el equipo directivo puede definir libremente la participación de los siguientes actores en el proceso de Evaluación Docente:
  - Autoevaluación.

- Evaluación del superior.
- Evaluación de docentes pares.
- Evaluación de Alumnos 7º - IV (P. Asig.-PJ).
- Evaluación de Apoderados (PJ-Educadora).

-Las respuestas de los cuestionarios se realiza en línea.

- El sistema garantiza soporte en línea durante 120 días.

- Una vez finalizado el proceso de evaluación, el director tendrá acceso a los informes de la evaluación a través de la página web del sistema. Se puede acceder a:

- Informes generales de la organización.
- Informes individuales de los docentes.
- Informes por ciclos de enseñanza.

- Confidencialidad: se accede a los cuestionarios e informes con una clave personal, de acuerdo al cargo.

### **1.2.2. Sistemas de gestión de la evaluación del desempeño en Cuba**

#### **Sistema de Evaluación del Desempeño del sector hotelero.**

El sistema permite, primeramente, diseñar para cada área o departamento el conjunto de indicadores generales y específicos con su peso de importancia, así como los grados para cada uno de ellos.

Para el logro de la competitividad deseada por la alta gerencia, los recursos humanos juegan un papel fundamental y por tanto un sistema de evaluación del desempeño (ED) que integre los objetivos individuales y colectivos con los de la alta gerencia se hace imprescindible. En este caso el sistema diseñado busca entre otros ese gran objetivo. De manera general el sistema introduce la automatización de la ED y una concepción más integral de este sistema enfocado al desarrollo del personal y de la entidad.

Globalmente permite:

- Que cada área diseñe sus propios indicadores de ED, en función de los objetivos trazados.
- La variación de los pesos específicos de los indicadores de ED para cada área en dependencia de las condiciones y objetivos de las mismas, haciendo al sistema flexible.
- Almacenar los sistemas de indicadores y pesos específicos de cada período evaluado para su posterior análisis y desarrollo.
- Una valoración cualitativa y cuantitativa de cada trabajador.
- Utilizar la valoración cuantitativa en los sistemas de estimulación y pago de idoneidad.
- Hacer proyecciones futuras para el desarrollo de los trabajadores a corto, mediano y largo plazo.
- Levantar necesidades de capacitación en dependencia de la evaluación de sus resultados.
- Rapidez y confiabilidad en la validación de los resultados de evaluación.
- Un modelo de salida con todos los resultados de cada indicador.
- Este programa está formado por dos sistemas que son: Editor de Evaluaciones y Evaluación del Desempeño.

El Editor de Evaluaciones permite diseñar el sistema de indicadores: en él se definen el conjunto de indicadores generales, indicadores específicos, grados, asumidos para cada indicador y peso específico de los mismos. Este sistema permite ser salvado y cargado posteriormente para el otro sistema de Evaluación del Desempeño. Esto quiere decir que para poder evaluar el desempeño de los trabajadores se hace necesario, primero, diseñar el sistema de indicadores a través del Editor de Evaluaciones o cargar uno similar de otro período que ya se tenga salvado y diseñado.

En el caso del sistema evaluador permite, luego de la entrada y verificación de los datos del individuo, dar una evaluación cualitativa para ser integrada y calculada posteriormente aplicando métodos de la matemática difusa.

Finalmente el programa muestra la evaluación de cada indicador por trabajador con su evaluación general cualitativa y cuantitativa (valor entre 0-1 después del proceso de defusificación) así como una vista gráfica del comportamiento de cada individuo.

### Modelos de Salida

En los modelos de salida se tratan de agrupar los datos que resultan de interés para los usuarios definiendo en cada uno de ellos un encabezamiento y cuerpo del modelo.

Los principales modelos de salida son los siguientes:

- Vista de Diseño de Evaluación: pertenece al sistema de Editor de Evaluaciones y muestra el diseño final del conjunto de indicadores (generales y específicos) con sus correspondientes grados y pesos específico.

Resultados finales: pertenece al sistema de Evaluación del Desempeño e incluye los siguientes modelos:

- Resultados finales de la evaluación por cada indicador con el resultado global (cualitativo y cuantitativo) para cada trabajador.
- Resultados finales por área o departamento que incluye a todos los trabajadores con el índice cualitativo y cuantitativo correspondiente.
- Resultados finales con salida a Excel. En él se incluyen todos los elementos de la evaluación para utilizarlo como base de datos y aprovechar las ventajas del paquete Excel.
- Vista gráfica de los resultados:
  - En ella se ofrece el comportamiento gráfico de la evaluación de cada trabajador para su mejor análisis, proyección y tendencia de sus trabajadores.
  - Una vez finalizada la etapa evaluativa se calculan los indicadores de auditoría con el fin de mantener una retroalimentación con otras áreas de la gestión de los recursos humanos y como mecanismo de diagnóstico.

Entre los principales indicadores definidos y calculados se encuentran: nivel de utilización del sistema, resultados obtenidos en la ED, evaluaciones discutidas con los trabajadores, Índice de conformidad con la ED, nivel de Feed-Back de los resultados y nivel de utilización para la selección. Los resultados de estos indicadores en las instalaciones que ya está implantado el sistema muestran la eficacia del mismo y el mejoramiento continuo de la calidad, el servicio y el desarrollo individual de los trabajadores.

### **Beneficios de la aplicación del sistema**

Para el estudio del beneficio y la repercusión del sistema diseñado con la ayuda de la aplicación del modelo matemático asistido por computadoras se toma como referencia la utilidad de la automatización, que mejora los beneficios que se pueden obtener.

Entre ellos tenemos:

- Velocidad de procesamiento de la información y obtención del resultado de la evaluación.
- Consulta de la información.
- Productividad del trabajo.
- Exactitud y consistencia de los datos.
- Interrelación de las áreas con el sistema.
- Seguridad de la información. (4)

### **1.3. Tendencias actuales**

#### **1.3.1. Software libre**

Es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Fundación de Software Libre, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie.

#### **1.4. Metodologías para el desarrollo de software**

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo es garantizar la eficacia y la eficiencia en el proceso de generación de software. Los riesgos a afrontar y los controles a establecer, varían en función de las diferentes



etapas del ciclo de vida de desarrollo. En la actualidad se han desarrollado tipos de métodos en el desarrollo del software que son los llamados métodos ligeros o ágiles y métodos pesados. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan lograr la calidad del software mediante el orden y la documentación, los métodos ágiles lo hacen por medio de una comunicación directa entre las personas que intervienen en el proceso. Dentro de estas metodologías las más populares y mundialmente utilizadas son: Las metodologías pesadas: METRICA 3 y Rational Unified Process (RUP) y la metodología ágil: Extreme Programming (XP).

### 1.4.1. Metodologías pesadas o altamente predictivas

#### Metodología Métrica 3

Esta metodología ofrece a las organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software dentro del marco que permite alcanzar los siguientes objetivos:

- Proporcionar o definir sistemas de información que ayuden a conseguir los fines de la organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- Dotar a la organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.
- Mejorar la productividad de los departamentos de sistemas y tecnologías de la información y las comunicaciones, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.
- Facilitar la operación, mantenimiento y uso de los productos de software obtenidos. En una única estructura la metodología MÉTRICA 3 cubre distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad.

La automatización de las actividades propuestas en la estructura de MÉTRICA 3 es posible ya que sus técnicas están soportadas por una amplia variedad de herramientas de ayuda al desarrollo disponibles en el mercado.

### **El Proceso Unificado de Rational**

El Proceso Unificado de Rational (RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto).

Fue desarrollado por Rational Software, y está integrado con todo el conjunto de herramientas del *Rational*. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación.

### **1.4.2. Metodologías ligeras**

#### **Scrum**

Es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nuevas funcionalidades. Las iteraciones en general tienen una duración entre 2 y 4 semanas. Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o *Extreme Programming*.

Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto que se está construyendo a las necesidades del

cliente. Se busca entregar un software que realmente resuelva las necesidades, aumentando la satisfacción del cliente. (5)

### **Extreme Programming**

Extreme Programming (XP) es la más destacada de los procesos ágiles de desarrollo de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto, es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. (5)

## **1.5. Lenguaje y herramientas para el modelado de software**

### **Lenguaje UML**

El Lenguaje Unificado de Modelación con su acrónimo en inglés Unified Modeling Language (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema. El UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

### **Visual Paradigm para UML**

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar

documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (5)

### 1.6. Tecnologías para la implementación de software

#### 1.6.1. Lenguajes utilizados en aplicaciones web

##### Lenguaje HTML

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML (acrónimo en inglés de *HyperText Markup Language*, en español Lenguaje de Marcas Hipertextuales). Es un lenguaje estático para el desarrollo de sitios web. Desarrollado por el *World Wide Web Consortium (W3C)*. Los archivos pueden tener las extensiones (htm, html).

##### Ventajas:

- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.
- Archivos pequeños.
- Despliegue rápido.

##### Desventajas:

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas.

### Lenguaje Javascript

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa *Netscape Communications*. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript.

El código Javascript puede ser integrado dentro de nuestras páginas web. Para evitar incompatibilidades el W3C diseñó un estándar denominado DOM (en inglés *Document Object Model*, en su traducción al español Modelo de Objetos del Documento).

#### Ventajas:

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código Javascript se ejecuta en el cliente.

#### Desventajas:

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés *Cross Site Scripting* renombrado a XSS por su similitud con las hojas de estilo CSS).

### Lenguaje PHP

Es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor*, (inicialmente se llamó *Personal Home Page*). Surgió en 1995, desarrollado por *PHP Group*.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP.

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

### Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

### Desventajas:

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación.

### Seguridad:

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

### Lenguaje Java Server Page

Es un lenguaje para la creación de sitios web dinámicos, JPS acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma, creado para ejecutarse del lado del servidor.

JSP fue desarrollado por *Sun Microsystems*. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los *servlets* de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

### Lenguaje Python

Es un lenguaje de programación creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.

Su código no necesita ser compilado, por lo que se llama que el código es interpretado. Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten por un estilo de programación particular:

- Programación orientada a objetos.
- Programación estructurada.
- Programación funcional.
- Programación orientada a aspectos.

### 1.6.2. Marcos de trabajo utilizados para el desarrollo web

#### **Symfony**

Symfony Project es un marco de trabajo para construir aplicaciones web con PHP. En otras palabras, Symfony es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web.

Para los iniciados en los marcos de trabajos, Symfony es una de las mejores copias para PHP del famoso marco de trabajo Ruby on Rails. Symfony ha tomado las mejores ideas de Rails y de muchos otros marcos de trabajos, ha incorporado ideas propias y el resultado es un marco de trabajo elegante, estable, productivo y muy bien documentado.

Symfony emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web. El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos.

La vista transforma la información obtenida por el modelo en las páginas web a las que acceden los usuarios. El controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows.

Symfony utiliza el formato YAML como el lenguaje preferido para almacenar su configuración porque es mucho más rápido de escribir que XML (ya que no hacen falta las etiquetas de cierre y el uso continuo de las comillas) y es mucho más poderoso que los tradicionales archivos `.ini`.

Symfony se diseñó para que se ajustara a los siguientes requisitos:



- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales, y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de *phpDocumentor* y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las librerías de otros fabricantes.

### Zend Framework

Zend Framework (ZF) es un marco de trabajo de código abierto para desarrollar aplicaciones web y servicios web con PHP5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible marco de trabajo de aplicaciones web al combinarse. ZF ofrece un gran rendimiento y una robusta implementación Modelo-Vista-Controlador (MVC), una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes, como `Zend_Auth` y `Zend_Acl`, proveen autenticación de usuarios y autorización diferentes a las tiendas de certificados comunes.

También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los servicios web más populares. Cualesquiera que sean las necesidades de su solicitud, se tienen todas las posibilidades de encontrar un componente de Zend Framework que se pueda utilizar para reducir drásticamente el tiempo de desarrollo, con una base completamente sólida. (6)

### **Seagull**

El marco de trabajo Seagull, nos permite realizar una programación modular, posee un Sistema de Gestión de Contenidos, con su acrónimo en inglés Content Management System (CMS). Entre sus características de encuentran su compatibilidad con PHP 4 y PHP 5, ORM integrado, incorpora el patrón MVC, uso de plantillas, soporte para múltiples bases de datos, validación de datos, alto nivel de configuración, autenticación, integración de librerías PEAR. (7)

### **Prado**

Prado está basado en componentes eventos con el objetivo de acelerar el desarrollo de aplicaciones web usando PHP 5.

El concepto del desarrollo de aplicaciones en Prado es diferente, se utilizan componentes, eventos y propiedades en vez de procedimientos, URL y parámetros.

Este marco de trabajo combina especificaciones en un archivo XML, plantillas HTML y una clase PHP. Prado, cuenta con soporte para AJAX, validación, autenticación, plantillas, múltiples bases de datos.

Entre los beneficios que podemos encontrar para el desarrollo de aplicaciones web se encuentran:

- Reutilización: los códigos y componentes pueden ser reutilizados.
- Fácil uso: la creación y uso de componentes es fácil.
- Funcionamiento: utiliza una técnica de caché para asegurar el funcionamiento de aplicaciones basadas en él.
- Integración: permite la separación del contenido y la presentación. (7)

### **1.6.3. Sistemas gestores de base de datos**

Los sistemas gestores de base de datos (SGBD), son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los mismos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los datos.

### MySQL

El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multi-threaded, multi usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido. MySQL es una marca registrada de MySQL AB.

El software MySQL tiene una doble licencia. Los usuarios pueden elegir entre usar el software MySQL como un producto de código libre bajo los términos de la licencia *GNU General Public License*.

### Oracle

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server, etc. (8)

### PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

## 1.7. Fundamentación metodológica

Después del estudio y el análisis de las metodologías, herramientas y tecnologías que se usan en el mundo actual en el desarrollo web. Después de valorar todas las posibilidades existentes, se concluye cuáles se adaptan a la propuesta de solución del sistema, concluyendo que:

Todas las herramientas y tecnologías son software libre, excepto una, de la cual la universidad posee su licencia.

Para guiar el proceso de desarrollo del software, se usa la metodología RUP, porque es muy sólida y robusta, la cual se utiliza como base para el desarrollo de este trabajo.

Para el modelado del negocio, modelado del sistema y análisis y diseño del sistema se utiliza el lenguaje de modelado UML y la herramienta para realizar el modelado sería *Visual Paradigm 6.1 Enterprise Edition*, debido a la adquisición por parte de la universidad de la licencia de la misma.

Para la implementación del sistema se propone la utilización del marco de trabajo *Symfony Project*, seleccionado por su gran adaptabilidad y usabilidad para aplicaciones empresariales, además de ser un marco de trabajo de código abierto.

### 1.8. Conclusiones

Después de realizar un análisis detallado de las tendencias actuales y las tecnologías que se usan actualmente en el mundo del software y específicamente en el desarrollo web se ha llegado a la conclusión de cuáles son las herramientas más adecuadas para darle solución a la problemática planteada. Es importante destacar que toda la tecnología que se ha decidido usar es completamente libre, excepto por una herramienta de la cual la universidad posee su licencia de uso.

Se estudiaron además sistemas de gestión de la evaluación del desempeño que son utilizados en algunos países del mundo así como en nuestro país por lo que se obtiene como resultado que ninguno de los sistemas anteriormente citados puede ser utilizado para la gestión de la evaluación del desempeño de los profesores en la facultad 1.

En el caso de EVAL 360 y QTraining se puede concluir en que ambos están orientados a la evaluación del desempeño en el ámbito empresarial, y no al desempeño de los trabajadores docentes. Otro de los inconvenientes que presentan estos sistemas es que están desarrollados sobre tecnologías propietarias lo que va en contra de las políticas de migración hacia software libre del país y la universidad.

En el caso del software desarrollado en México no está disponible en el mercado, ya que fue creado para una red nacional del país, además de que no se tiene información suficiente de sus funcionalidades así como de pruebas que den una opinión favorable de su experiencia como sistema.

En el caso del software desarrollado en Cuba, no contempla las necesidades específicas de la evaluación del desempeño de los profesores.

## CAPÍTULO 2

### CARACTERÍSTICAS DEL SISTEMA

#### 2.1. Introducción

En este capítulo se desea hacer una caracterización acerca del sistema que se va a realizar, es decir, la propuesta de sistema, el modelo de casos de uso del negocio con la descripción general de los procesos de negocio propuestos, incluyendo los actores y trabajadores que participan en el mismo. Además se especificarán los principales requisitos del software, mediante los requerimientos funcionales y no funcionales, dando paso así a un buen modelado del sistema.

#### 2.2. Principales reglas del negocio

Reglas del negocio que se deben tener en cuenta para modelar o automatizar el proceso de desarrollo:

- Los profesores solo están autorizados a enviar un plan de trabajo mensual.
- Para realizar cambios en este plan de trabajo, el profesor está obligado a realizar una solicitud a su jefe de departamento.
- Los profesores deben estar previamente inscritos en el sitio, para poder realizar cualquier acción.
- Para verificar que la información enviada fue aprobada el profesor debe revisar su plan de trabajo periódicamente.
- Los usuarios tendrán acceso limitado en correspondencia con el rol que se le asigne.

#### 2.3. Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. (9)

Nombre del Actor	Descripción
Evaluado	Es el encargado de insertar las tareas a su plan de trabajo.

Tabla 2.1. Descripción de actores del negocio.

## 2.4. Trabajadores del negocio

Un trabajador del negocio representa a un ser humano, software o hardware que desempeña un rol dentro de las Realizaciones del Caso de Uso del Negocio. Este trabajador interactúa con entidades y otros trabajadores para que el negocio funcione. Los trabajadores de negocio son roles y no posiciones organizacionales, ya que una persona puede desempeñar varios roles pero sólo tiene una posición en la organización. (9)

Nombre del Trabajador	Descripción
Evaluador	Son los encargados de modificar las tareas del plan de trabajo de cada trabajador. Es responsable emitir una evaluación sobre el plan de trabajo.
Consultor	Consulta las evaluaciones realizadas a los trabajadores cuando sea necesario.

Tabla 2.2. Descripción de trabajadores del negocio.

## 2.5. Diagrama de casos de uso del negocio

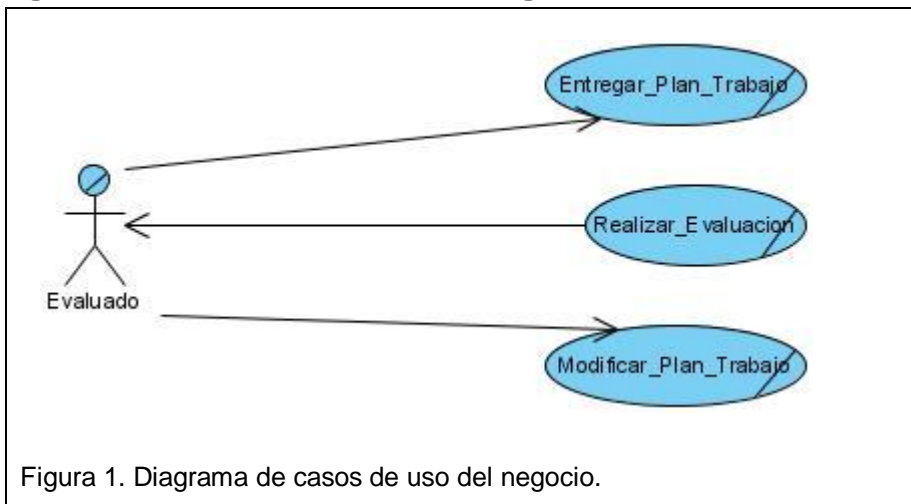


Figura 1. Diagrama de casos de uso del negocio.

## 2.6. Descripción textual de los casos de uso del negocio

A continuación se muestra una descripción detallada de los casos de uso del negocio, los diagramas de actividades correspondientes pueden ser encontrados en el Anexo 1.

Caso de Uso	Entregar Plan de Trabajo	
Actores	Evaluado (Inicia)	
Trabajadores	Evaluador	
Resumen	Este caso de uso se inicia cuando el profesor inserta las tareas a realizar en su plan de trabajo.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Propone Plan de Trabajo.	1.1 Se recibe la propuesta de Plan de Trabajo. 1.2 Se procede a Revisión de Plan de Trabajo. 1.3 Se revisa la propuesta de Plan de Trabajo. 1.4 Se aprueba Plan de Trabajo	
Flujo Alternativo 1.3		
	1.3 No se aprueba la propuesta de Plan de Trabajo.	

Tabla 2.3: Descripción textual del caso de uso “Entregar Plan de Trabajo”.

Caso de Uso	Modificar Plan de Trabajo	
Actores	Evaluado	
Trabajadores	Evaluador	
Resumen	Este caso de uso se inicia cuando el evaluado realiza una solicitud a su evaluador inmediato para realizar cambios en su plan de trabajo inicial y este procede a realizar la acción.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Realiza una solicitud de cambio en el plan de trabajo.	1.1 Se atiende solicitud. 1.2 Se modifica plan de trabajo. 1.3 Se informa modificación.	
Flujo Alterno 1.1		
	1.1 Sino acepta la solicitud, deniega la solicitud recibida.	

Tabla 2.4. Descripción textual del caso de uso “Modificar Plan de Trabajo”.



Caso de Uso	Realizar Evaluación	
Actores	Evaluado	
Trabajadores	Evaluador	
Resumen	Este caso de uso se inicia cuando están insertadas todas las tareas del plan de trabajo por parte del profesor y el jefe de departamento procede a realizar dicho trabajo.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
Informa inserción del plan de trabajo.	1.1 Se recibe información. 1.2 Se realiza revisión. 1.3 Aprueba Plan de Trabajo. 1.4 Se realiza una evaluación de las tareas del plan de trabajo.	
Flujo Alterno 1.2		
	1.2 Si la revisión no es satisfactoria, se deniega el plan de trabajo.	

Tabla 2.5. Descripción textual del caso de uso “Realizar Evaluación”.

## 2.7. Requerimientos funcionales

### R1.Gestionar evaluación

#### 1.1. Adicionar evaluación

- 1.1.1. El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al administrador de crear una evaluación.
- 1.1.2. Se validan los datos entrados.
  - 1.1.2.1. Si existen campos vacíos se notifica.
- 1.1.3. Se almacena la evaluación adicionada
- 1.1.4. El sistema debe notificar que se creó la evaluación.

#### 1.2. Modificar evaluación

- 1.2.1. El sistema debe mostrar un formulario con todas las evaluaciones creadas.

- 1.2.2. El sistema debe mostrar un formulario con todos los datos modificables.
  - 1.2.2.1. Si se modifican los datos el sistema debe notificarlo.
- 1.2.3. Se almacenan los datos.
- 1.3. Eliminar evaluación
  - 1.3.1. El sistema debe permitir eliminar la evaluación seleccionada.
    - 1.3.1.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.
  - 1.3.2. El sistema debe notificar que la evaluación se eliminó satisfactoriamente.
- 1.4. Mostrar evaluación
  - 1.4.1. El sistema debe mostrar un formulario con todas las evaluaciones que se hayan creado.
    - 1.4.1.1. Si no existen evaluaciones, el sistema debe mostrar un mensaje notificando que el listado está vacío.

### **R2. Gestionar área**

- 2.1. Adicionar área
  - 2.1.1. El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al administrador de crear un área.
  - 2.1.2. Se validan los datos entrados.
    - 2.1.2.1. Si existen campos vacíos se notifica.
  - 2.1.3. Se almacena el área adicionada.
  - 2.1.4. El sistema debe notificar que se creó el área.
- 2.2. Modificar área
  - 2.2.1. El sistema debe mostrar un formulario con todas las áreas creadas.
  - 2.2.2. El sistema debe mostrar un formulario con todos los datos modificables.
    - 2.2.2.1. Si se modifican los datos el sistema debe notificarlo.
  - 2.2.3. Se almacenan los datos.

- 2.3. Eliminar área
  - 2.3.1. El sistema debe permitir eliminar el rol seleccionado
    - 2.3.1.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.
  - 2.3.2. El sistema debe notificar que el área se eliminó satisfactoriamente.
- 2.4. Mostrar área
  - 2.4.1. El sistema debe mostrar un formulario con todas las áreas que se hayan creado.
    - 2.4.1.1. Si no existen áreas, el sistema debe mostrar un mensaje notificando que el listado está vacío.

### **R3. Gestionar cargo**

- 3.1. Adicionar cargo
  - 3.1.1. El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al administrador de crear un cargo.
  - 3.1.2. Se validan los datos entrados.
  - 3.1.3. Se almacena el cargo adicionado
    - 3.1.3.1. Si el cargo ya existe se notifica y se da la posibilidad de crear el cargo nuevamente.
  - 3.1.4. El sistema debe notificar que se creó el cargo.
- 3.2. Modificar cargo
  - 3.2.1. El sistema debe mostrar un formulario con todos los cargos creados.
  - 3.2.2. El sistema debe mostrar un formulario con todos los datos modificables.
    - 3.2.2.1. Si se modifican los datos el sistema debe notificarlo.
  - 3.2.3. Se almacenan los datos.
- 3.3. Eliminar cargo
  - 3.3.1. El sistema debe permitir eliminar el cargo seleccionado
    - 3.3.1.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.
  - 3.3.2. El sistema debe notificar que el cargo se eliminó satisfactoriamente.

### 3.4. Mostrar cargo

3.4.1. El sistema debe mostrar un formulario con todos los cargos que se hayan creado.

3.4.1.1. Si no existen cargos, el sistema debe mostrar un mensaje notificando que el listado está vacío.

## R4. Gestionar usuario

### 4.1. Adicionar usuario

4.1.1. El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al administrador de crear un usuario con dominio.

4.1.2. Se validan los datos entrados.

4.1.2.1. Si existen campos vacíos se notifica.

4.1.3. Se almacena el usuario adicionado

4.1.3.1. Si el usuario ya existe se notifica y se da la posibilidad de crear el usuario nuevamente.

4.1.4. El sistema debe notificar que se creó el usuario.

### 4.2. Modificar usuario

4.2.1. El sistema debe mostrar un formulario con todos los usuarios creados.

4.2.1.1. Si no existen usuarios en el sistema no se permiten cambios.

4.2.2. El sistema debe mostrar un formulario con todos los datos modificables.

4.2.2.1. Si se modifican los datos el sistema debe notificarlo.

4.2.3. Se almacenan los datos.

### 4.3. Eliminar usuario

4.3.1. El sistema debe permitir eliminar el usuario seleccionado

4.3.1.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.

4.3.2. El sistema debe notificar que el usuario se eliminó satisfactoriamente.

### 4.4. Mostrar usuario

4.4.1. El sistema debe mostrar un formulario con todos los usuarios que se hayan creado.

- 4.4.1.1. Si no existen usuarios, el sistema debe mostrar un mensaje notificando que el listado está vacío.

### **R5. Crear cuenta local**

- 5.1. El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al usuario de crear una cuenta local.
- 5.2. Se validan los datos.
  - 5.2.1. Si el usuario ya existe se notifica y se da la posibilidad de crear el usuario nuevamente.
- 5.3. Se almacenan los datos.

### **R6. Gestionar tipo de tarea**

- 6.1. Adicionar tipo de tarea
  - 6.1.1. El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al administrador de crear un tipo de tarea.
  - 6.1.2. Se validan los datos entrados.
    - 6.1.2.1. Si existen campos vacíos se notifica.
  - 6.1.3. Se almacena el tipo de tarea adicionado.
  - 6.1.4. El sistema debe notificar que se creó el tipo de tarea.
- 6.2. Modificar tipo de tarea
  - 6.2.1. El sistema debe mostrar un formulario con todos los tipos de tarea creados.
    - 6.2.1.1. Si no existen tipos de tarea en el sistema, no se permiten cambios.
  - 6.2.2. El sistema debe mostrar un formulario con todos los datos modificables.
    - 6.2.2.1. Si se modifican los datos el sistema debe notificarlo.
  - 6.2.3. Se almacenan los datos.
- 6.3. Eliminar tipo de tarea
  - 6.3.1. El sistema debe permitir eliminar el tipo de tarea seleccionado
    - 6.3.1.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.
  - 6.3.2. El sistema debe notificar que el tipo de tarea se eliminó satisfactoriamente.

### 6.4. Mostrar tipo de tarea

6.4.1. El sistema debe mostrar un formulario con todos los tipos de tareas que se hayan creado.

6.4.1.1. Si no existen tipos de tareas, el sistema debe mostrar un mensaje notificando que el listado está vacío.

## R7. Gestionar parámetro

### 7.1. Adicionar parámetro

7.1.1. El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al administrador de crear un parámetro.

7.1.2. Se validan los datos entrados.

7.1.2.1. Si existen campos vacíos se notifica.

7.1.3. Se almacena el parámetro adicionado.

7.1.4. El sistema debe notificar que se creó el parámetro.

### 7.2. Modificar parámetro

7.2.1. El sistema debe mostrar un formulario con todos los parámetros creados.

7.2.1.1. Si no existen parámetros en el sistema, no se permiten cambios.

7.2.2. El sistema debe mostrar un formulario con todos los datos modificables.

7.2.2.1. Si se modifican los datos el sistema debe notificarlo.

7.2.3. Se almacenan los datos.

### 7.3. Eliminar parámetro

7.3.1. El sistema debe permitir eliminar el parámetro seleccionado.

7.3.1.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.

7.3.2. El sistema debe notificar que el parámetro se eliminó satisfactoriamente.

### 7.4. Mostrar parámetro

7.4.1. El sistema debe mostrar un formulario con todos los parámetros que se hayan creado.

7.4.1.1. Si no existen parámetros, el sistema debe mostrar un mensaje notificando que el listado está vacío.

### **R8. Gestionar tarea evaluada**

#### 8.1. Adicionar tarea evaluada

8.1.1. El sistema debe dar la posibilidad al administrador de crear una tarea evaluada partiendo de las tareas asignadas a los cargos, a las áreas, a los cargos de un área específica.

8.1.2. Se verifican los datos.

8.1.2.1. Si hay campos vacíos se notifica.

8.1.3. Se almacena la tarea evaluada adicionada.

8.1.4. El sistema debe notificar que se creó la tarea evaluada.

#### 8.2. Modificar tarea evaluada

8.2.1. El sistema debe mostrar un formulario con todas las tareas evaluadas creadas.

8.2.2. El sistema debe mostrar un formulario con todos los datos modificables.

8.2.2.1. Si se modifican los datos el sistema debe notificarlo.

8.2.3. Se almacenan los datos.

#### 8.3. Eliminar tarea evaluada

8.3.1. El sistema debe permitir eliminar la tarea evaluada seleccionada.

8.3.1.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.

8.3.2. El sistema debe notificar que la tarea evaluada se eliminó satisfactoriamente.

#### 8.4. Mostrar tarea evaluada

8.4.1 El sistema debe mostrar un formulario con todas las tareas evaluadas que se hayan creado.

8.4.1.1. Si no existen tareas evaluadas, el sistema debe mostrar un mensaje notificando que el listado está vacío.

### **R9. Gestionar certificado**

#### 9.1. Adicionar certificado

- 9.1.1. El sistema debe mostrar todas las tareas que tenga un usuario y debe dar la posibilidad al evaluador de crear un certificado.
  - 9.1.1.1. Si al menos una de las tareas no está evaluada, se notifica.
- 9.1.2. Se validan los datos.
  - 9.1.2.1. Si existen campos vacíos, se notifica.
- 9.1.3. Se guardan los datos.
- 9.1.4. El sistema debe notificar que se guardaron los datos.

### 9.2. Modificar certificado

- 9.2.1. El sistema debe mostrar un formulario con los certificados creados.
- 9.2.2. El sistema debe mostrar un formulario con todos los datos modificables.
  - 9.2.2.1. Si se modifican los datos el sistema debe notificarlo.
- 9.2.3. Se almacenan los datos y se notifica.

### 9.3. Eliminar certificado

- 9.3.1. El sistema debe permitir eliminar el certificado seleccionado.
  - 9.3.1.1. Si el usuario no es el evaluador definidor, solo puede eliminar el certificado que él haya creado.
- 9.3.2. El sistema debe mostrar un mensaje de confirmación antes de eliminar.
- 9.3.3. El sistema debe notificar que el certificado se eliminó satisfactoriamente.

### 9.4. Mostrar certificado

- 9.4.1. El sistema debe mostrar un formulario con todos los certificados que se hayan creado.
  - 9.4.1.1. Si no existen certificados, el sistema debe mostrar un mensaje notificando que el listado está vacío.

## **R10. Gestionar tipo de evaluación**

### 10.1. Adicionar tipo de evaluación

- 10.1.1 El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al administrador de crear un tipo de evaluación.



10.1.2. Se validan los datos entrados.

10.1.2.1. Si existen campos vacíos se notifica.

10.1.3. Se almacena el tipo de evaluación adicionado

10.1.4. El sistema debe notificar que se creó el tipo de evaluación.

10.2. Modificar tipo de evaluación

10.2.2. El sistema debe mostrar un formulario con todos los tipos de evaluación creados.

10.2.2.1. Si no existen tipos de evaluación en el sistema, no se permiten cambios.

10.2.3. El sistema debe mostrar un formulario con todos los datos modificables.

10.2.3.1. Si se modifican los datos el sistema debe notificarlo.

10.2.4. Se almacenan los datos.

10.3. Eliminar tipo de evaluación

10.3.2. El sistema debe permitir eliminar el tipo de evaluación seleccionado

10.3.2.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.

10.3.3. El sistema debe notificar que el tipo de evaluación se eliminó satisfactoriamente.

10.4. Mostrar tipos de evaluación

10.4.2. El sistema debe mostrar un formulario con todos los tipos de evaluación que se hayan creado.

10.4.2.1. Si no existen tipos de evaluación, el sistema debe mostrar un mensaje notificando que el listado está vacío.

### **R11. Gestionar tarea**

11.1. Adicionar tarea

11.1.1. El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al usuario de crear una tarea.

11.1.2. Se validan los datos entrados.

11.1.2.1. Si existen campos vacíos se notifica.

11.1.3. Se almacena la tarea adicionada

11.1.4. El sistema debe notificar que se creó la tarea.

11.2. Modificar tarea

11.2.1. El sistema debe mostrar un formulario con todas las tareas creadas.

11.2.1.1 Si no existen tareas en el sistema, no se permiten cambios.

11.2.2. El sistema debe mostrar un formulario con todos los datos modificables.

11.2.2.1. Si se modifican los datos el sistema debe notificarlo.

11.2.3. Se almacenan los datos.

11.3. Eliminar tarea

11.3.1. El sistema debe permitir eliminar la tarea seleccionada.

11.3.1.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.

11.3.2. El sistema debe notificar que la tarea se eliminó satisfactoriamente.

11.4. Mostrar tarea

11.4.1. El sistema debe mostrar un formulario con todas las tareas que se hayan creado.

11.4.1.1. Si no existen tareas, el sistema debe mostrar un mensaje notificando que el listado está vacío.

**R12. Gestionar tarea especial**

12.1. Adicionar tarea especial

12.1.1. El sistema debe mostrar un listado con las tareas evaluadas y para cada una los aspectos que se le pueden adicionar y debe dar la posibilidad al administrador de crear una tarea especial.

12.1.2. Se almacena la tarea especial adicionada.

12.1.3. El sistema debe notificar que se guardaron los cambios.

12.2. Modificar tarea especial

12.2.1. El sistema debe mostrar un formulario con todas las tareas especiales creadas.

12.2.2. El sistema debe mostrar un formulario con todos los datos modificables.

12.2.2.1. Si se modifican los datos el sistema debe notificarlo.

12.2.3. Se almacenan los datos.

12.3. Eliminar tarea especial

12.3.1. El sistema debe permitir eliminar la tarea especial seleccionada.

12.3.1.1. El sistema debe mostrar un mensaje de confirmación antes de eliminar.

12.3.2. El sistema debe notificar que la tarea especial se eliminó satisfactoriamente.

12.4. Mostrar tarea especial

12.4.1. El sistema debe mostrar un formulario con todas las tareas especiales que se hayan creado.

12.4.1.1. Si no existen tareas especiales, el sistema debe mostrar en cada tarea evaluada que no tiene aspectos.

### **R13. Asignar tareas**

13.1. Asignar tareas a áreas

13.1.1. El sistema debe mostrar las asignaciones existentes.

13.1.1.1. Si no existen tareas o áreas, se muestra una notificación.

13.1.2. El sistema debe permitir que el usuario seleccione las tareas que quiere asignar a un área.

13.1.3. Se guardan los datos.

13.1.4. Se notifica que se guardaron los datos.

13.2. Asignar tareas a cargos

13.2.1. El sistema debe mostrar las asignaciones existentes.

13.2.1.1. Si no existen tareas o cargos, se muestra una notificación.

13.2.2. El sistema debe permitir que el usuario seleccione las tareas que quiere asignar a un cargo.

13.2.3. Se guardan los datos.

13.2.4. Se notifica que se guardaron los datos.

13.3. Asignar tareas a usuarios

13.3.1. El sistema debe mostrar las asignaciones existentes.

13.3.1.1. Si no existen tareas o usuarios, se muestra una notificación.

13.3.2. El sistema debe permitir que el usuario seleccione las tareas que quiere asignar a un usuario.

13.3.3. Se guardan los datos.

13.3.4. Se notifica que se guardaron los datos.

13.4. Asignar tareas a cargos de un área específica

13.4.1. El sistema debe mostrar las asignaciones existentes.

13.4.1.1. Si no existen tareas o cargos asignados a las áreas se muestra una notificación.

13.4.2. El sistema debe permitir que el usuario seleccione las tareas que quiere asignar a los cargos de un área en específico.

13.4.3. Se guardan los datos.

13.4.4. Se notifica que se guardaron los datos.

**R14. Asignar funcionalidades**

14.1. El sistema debe mostrar las asignaciones existentes.

14.1.1. Si no existen cargos, se muestra una notificación.

14.2. El sistema debe permitir que el usuario seleccione las funcionalidades que quiere asignar a un cargo.

14.3. Se guardan los datos.

14.4. Se notifica que se guardaron los datos.

**R15. Asignar cargos**

15.1. Asignar cargos a usuario

15.1.1. El sistema debe mostrar las asignaciones existentes.

15.1.1.1. Si no existen usuarios o cargos, se muestra una notificación.

15.1.2. El sistema debe permitir que el usuario seleccione los cargos que quiere asignar a un usuario en específico.

15.1.3. Se guardan los datos.

15.1.4. Se notifica que se guardaron los datos.

15.2. Asignar cargos a área.

15.2.1. El sistema debe mostrar las asignaciones existentes.

15.2.1.1. Si no existen áreas o cargos, se muestra una notificación.

15.2.2. El sistema debe permitir que el usuario seleccione los cargos que quiere asignar a un área en específico.

15.2.3. Se guardan los datos.

15.2.4. Se notifica que se guardaron los datos.

### **R16. Asignar áreas**

16.1. El sistema debe mostrar las asignaciones existentes.

16.1.1. Si no existen áreas o usuarios, se muestra una notificación.

16.2. El sistema debe permitir que el usuario seleccione las áreas que quiere asignar a un usuario en específico.

16.3. Se guardan los datos.

16.4. Se notifica que se guardaron los datos.

### **R17. Asignar cargo evaluado**

17.1. El sistema debe mostrar las asignaciones existentes.

17.1.1. Si no existen cargos asignados a las áreas, se muestra una notificación.

17.2. El sistema debe permitir que el usuario seleccione los cargos que van a ser evaluados por un cargo evaluador, sin importar el área a que pertenezcan.

17.3. Se guardan los datos.

17.4. Se notifica que se guardaron los datos.

### **R18. Autenticar Usuario**

18.1. El sistema debe mostrar a los usuarios la pantalla de autenticación

18.2. El sistema debe mostrar los campos: usuario y contraseña.

18.3. Se validan los datos entrados.

18.3.1. Si el usuario no posee privilegios para acceder, el sistema debe notificarlo e impedirle el acceso.

18.4. El sistema debe mostrar las opciones a las que el usuario tiene acceso.

## 2.8. Requerimientos no funcionales

Entre los requisitos no funcionales definidos para el sistema propuesto se encuentran:

**Apariencia de Interfaz o Externa:** El sistema contará con una interfaz sencilla y fácil de entender por los usuarios.

**Usabilidad:** El sistema podrá ser utilizado por personas que posean un conocimiento básico de computación.

**Rendimiento:** El sistema trabajará con grandes volúmenes de información, ya que contará con la información de todos los trabajadores de la facultad.

**Soporte:** El sistema permitirá que se le realicen cambios futuros en caso de que cambien las necesidades del cliente.

**Portabilidad:** El sistema funcionará en cualquier equipo con Microsoft Windows 98 o posterior.

**Seguridad:** El sistema cuenta con conjunto de roles que hacen más segura la aplicación restringiendo los permisos de cada usuario. Además de que para realizar alguna acción en la aplicación el usuario tiene que estar previamente autenticado.

### Software:

El sistema en el servidor deberá estar soportado en el Wamp5 1.7.4

El Gestor de Base de Datos que se debe utilizar es PostgreSQL 8.2

## 2.9. Actores del sistema

Actores	Descripción
Evaluador	Son los encargados de modificar las tareas del plan de trabajo de cada trabajador. Es responsable emitir una evaluación sobre el plan de trabajo.
Administrador	Tiene control total en el sistema y es el encargado de gestionar todos los datos que existen en la base de datos.

Tabla 2.6. Actores del sistema.

2.10. Diagrama de casos de uso del sistema

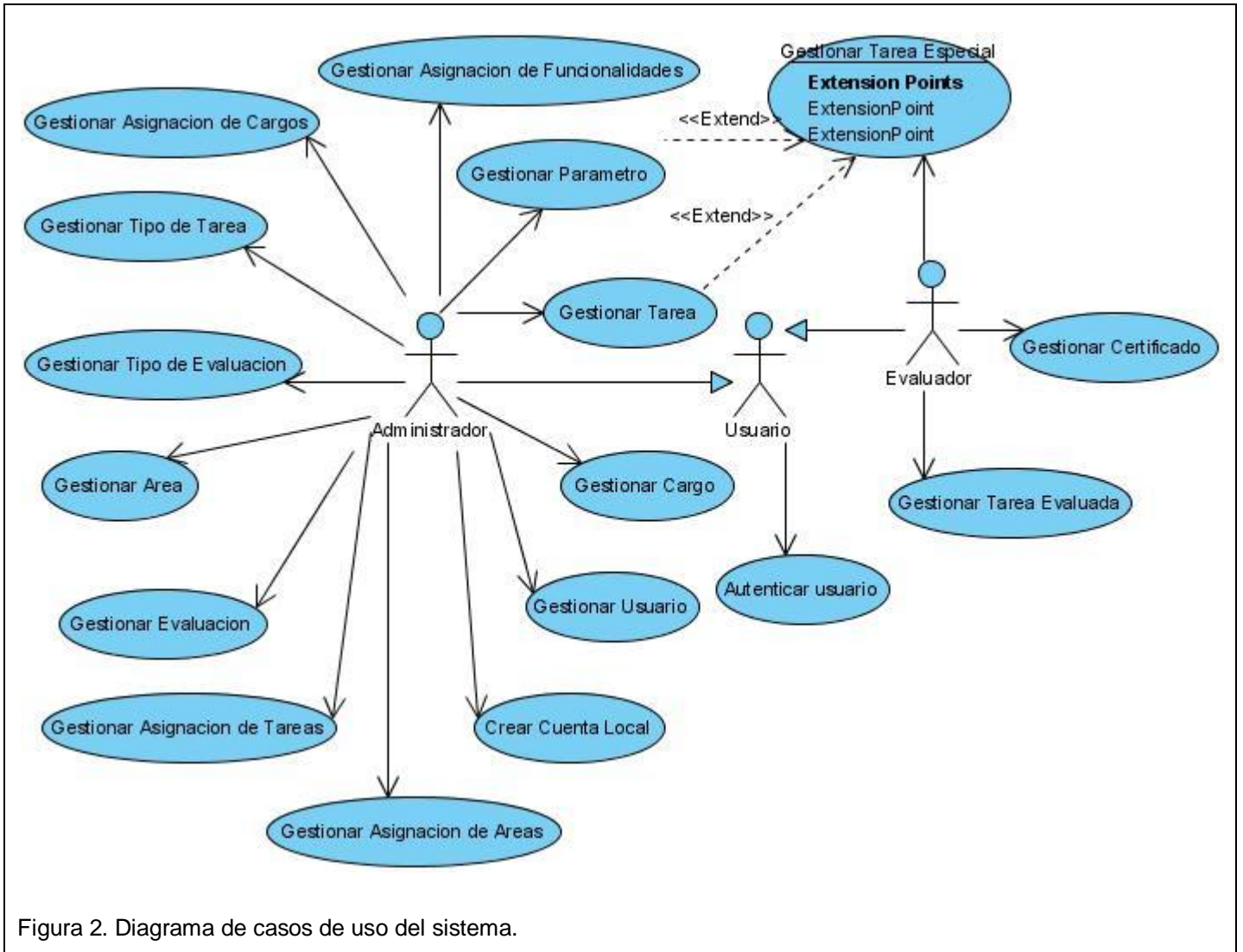


Figura 2. Diagrama de casos de uso del sistema.

### 2.11. Descripción abreviada de los casos de uso del sistema

A continuación se muestra una descripción abreviada de los casos de uso del sistema.

CU_1	Gestionar Evaluación
Actor	Administrador
Descripción	El caso de uso se inicia cuando el administrador solicita insertar una evaluación, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.
Referencia	R1

Tabla 2.7. Resumen del caso de uso “Gestionar evaluación”.

CU_2	Gestionar Área
Actor	Administrador
Descripción	El caso de uso se inicia cuando el administrador solicita insertar un área, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.
Referencia	R2

Tabla 2.8. Resumen del caso de uso “Gestionar área”.

CU_3	Gestionar Cargo
Actor	Administrador
Descripción	El caso de uso se inicia cuando el administrador solicita insertar un cargo, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.
Referencia	R3

Tabla 2.9. Resumen del caso de uso “Gestionar cargo”.



CU_4	Gestionar Usuario
Actor	Administrador
Descripción	El caso de uso se inicia cuando el administrador solicita insertar un cargo, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.
Referencia	R4

Tabla 2.10. Resumen del caso de uso “Gestionar usuario”.

CU_5	Gestionar Usuario Local
Actor	Administrador
Descripción	El caso de uso se inicia cuando un usuario indiferentemente del rol que posea, desea una cuenta local, es decir que no está dentro del dominio de la universidad, es una opción que se le da al usuario.
Referencia	R5

Tabla 2.11. Resumen del caso de uso “Gestionar usuario local”.

CU_6	Gestionar Tipo de Tarea
Actor	Administrador
Descripción	El administrador solicita gestionar tipos de tareas, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.
Referencia	R6

Tabla 2.12. Resumen del caso de uso “Gestionar tipo de tarea”.

CU_7	Gestionar Parámetro
Actor	Administrador
Descripción	El administrador solicita gestionar parámetros, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción a realizar.
Referencia	R7

Tabla 2.13. Resumen del caso de uso “Gestionar parámetro”.

CU_8	Gestionar Tarea Evaluada
Actor	Evaluador
Descripción	El evaluado solicita gestionar las tareas evaluadas de un usuario, el sistema le muestra los listados de tareas y de parámetros del mismo, además de las opciones de gestión a realizar.
Referencia	<ol style="list-style-type: none"> <li>1. R7</li> <li>2. R8</li> <li>3. R11</li> </ol>

Tabla 2.14. Resumen del caso de uso "Gestionar tarea evaluada".

CU_9	Gestionar Certificado
Actor	Evaluador
Descripción	El evaluador solicita gestionar el certificado de evaluación de un usuario, el sistema, le muestra todas las tareas evaluadas (algunas de estas tareas pueden ser especiales) que éste pueda tener, sin importar quien la haya evaluado.
Referencia	R9

Tabla 2.15. Resumen del caso de uso "Gestionar certificado".

CU_10	Gestionar Tipo de Evaluación
Actor	Administrador
Descripción	El administrador solicita gestionar tipos de evaluación, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción a realizar.
Referencia	R10

Tabla 2.16. Resumen del caso de uso "Gestionar tipo de evaluación".

CU_11	Gestionar tarea
Actor	Administrador, Evaluador, Evaluado
Descripción	El evaluador solicita gestionar tareas, el sistema muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción a realizar.
Referencia	R11

Tabla 2.17. Resumen del caso de uso "Gestionar tarea".

CU_12	Gestionar Tarea Especial
Actor	Evaluador
Descripción	El administrador solicita gestionar tipos de evaluación, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción a realizar.
Referencia	R12

Tabla 2.18. Resumen del caso de uso “Gestionar tareas especiales”.

CU_13	Asignar tareas
Actor	Administrador
Descripción	El administrador solicita asignar tareas, el sistema le permite asignarle tareas a un cargo, un área o un cargo de un área en específico.
Referencia	R13

Tabla 2.19. Resumen del caso de uso “Asignar Tareas”.

CU_14	Asignar funcionalidades
Actor	Administrador
Descripción	El administrador solicita asignar funcionalidades, el sistema le permite asignarle las funcionalidades pertinentes al usuario seleccionado.
Referencia	R14

Tabla 2.20. Resumen del caso de uso “Asignar funcionalidades”.

CU_15	Asignar cargos
Actor	Administrador
Descripción	El administrador solicita asignar cargos, el sistema le permite asignarle cargos a un usuario o un área.
Referencia	R15

Tabla 2.21. Resumen del caso de uso “Asignar cargos”.

CU_16	Asignar áreas
Actor	Administrador
Descripción	El administrador solicita asignar áreas, el sistema permite que se asignen áreas a los usuarios que se encuentren insertados en la base de datos.
Referencia	R16

Tabla 2.22. Resumen del caso de uso "Asignar áreas".

CU_17	Asignar cargo evaluado
Actor	Administrador
Descripción	El administrador solicita asignar los cargos evaluados. El sistema le permite seleccionar los cargos que van a ser evaluados por un cargo específico.
Referencia	R17

Tabla 2.23. Resumen del caso de uso "Asignar cargo evaluado".

CU_19	Autenticar usuario
Actor	Administrador Evaluador Evaluado
Descripción	El usuario solicita autenticarse, el sistema le muestra la pantalla de autenticación.
Referencia	R18

Tabla 2.24. Resumen del caso de uso "Autenticar usuario".

## 2.12. Estimación de esfuerzo

La planificación es una actividad de gran importancia en el desarrollo de software, al establecerse los objetivos y metas del sistema por desarrollar, a la vez que ayuda a valorar sus costos. Para que la planificación se logre efectuar de una forma eficiente, resulta fundamental evaluar el sistema de software por desarrollar, con el fin de estimar su nivel de dificultad, buscando obtener un aproximado del tiempo que será requerido en el desarrollo del mismo. Además, permite determinar el esfuerzo o

bien la cantidad de personas necesarias, para lograr finalizar dicho proyecto en un tiempo establecido.

**Método de Estimación por Puntos de Caso de Uso**

Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

**1. Cálculo de peso de los casos de uso sin ajustar**

Se calcula a partir de la siguiente ecuación: **UUCP = UAW + UUCW**

Teniendo en cuenta que:

**UUCP:** Puntos de Casos de Uso sin ajustar

**UAW:** Factor de Peso de los Actores sin ajustar

**UUCW:** Factor de Peso de los Casos de Uso sin ajustar

**Factor de Peso de los Actores sin ajustar (UAW)**

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Actor	Descripción	Complejidad	Factor de peso
Administrador del Sistema	Persona que interactúa con el sistema mediante una interfaz gráfica.	Complejo	3
Evaluador			

Tabla 2.25. Factor de peso de los casos de uso sin ajustar.

**UAW:** sumatoria de la cantidad de actores de un tipo de complejidad con su factor de peso.

Cantidad de actores de tipo complejo: 2

$$UAW = \sum (\text{actores} * \text{Peso})$$

$$UAW = 2 * 3$$

$$UAW = 6$$

**Factor de Peso de los Casos de Uso sin ajustar (UUCW)**

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo. Una transacción se entiende como una secuencia de actividades atómica, es decir, está representada por uno o más pasos del flujo de eventos principal del Caso de Uso, pudiendo existir más de una transacción dentro del mismo Caso de Uso.

Caso de Uso	Descripción	Complejidad	Factor de peso
Autenticar usuario	Contiene 1 transacción	3	5
Gestionar Evaluación	Contiene 4 transacciones		
Gestionar Área	Contiene 4 transacciones		
Gestionar Cargo	Contiene 4 transacciones		
Gestionar Usuario	Contiene 4 transacciones		
Crear Cuenta Local	Contiene 1 transacciones		
Gestionar Tipo de Tarea	Contiene 4 transacciones		
Gestionar Parámetro	Contiene 4 transacciones		
Gestionar Tarea Evaluada	Contiene 4 transacciones		
Gestionar Certificado	Contiene 4 transacciones		
Gestionar Tipo de Evaluación	Contiene 4 transacciones		
Gestionar Tarea	Contiene 4 transacciones		
Gestionar Tarea Especial	Contiene 4 transacciones		
Gestionar Asignación de Tarea	Contiene 4 transacciones		
Gestionar Asignación de Funcionalidad	Contiene 1 transacciones		
Gestionar Asignación de Cargo	Contiene 2 transacciones		
Gestionar Asignación de Área	Contiene 1 transacciones		

Tabla 2.26. Factor de peso de los casos de uso sin ajustar.

**UUCW:** Casos de uso del sistema y su complejidad.

$$UUCW = \sum CU * \text{Peso}$$

$$UUCW = 18 * 5$$

$$UUCW = 90$$

Finalmente los puntos de casos de uso sin ajustar resultan:

$$UUCP = UAW + UUCW$$

$$UUCP = 6 + 90$$

$$UUCP = 96$$

## 2. Cálculo de peso de los casos de uso ajustados

$$UCP = UUCP * TCF * EF$$

Teniendo en cuenta que:

UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor ambiente.

### Factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Factor	Descripción	Peso	Valor asignado	Peso * Valor
T1	Sistema distribuido	2	0	0
T2	Objetivos de performance o tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	2	2

T4	Procesamiento interno complejo	1	2	2
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	3	1.5
T7	Facilidad de uso	0.5	3	1.5
T8	Portabilidad	2	3	6
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	2	2

Tabla 2.27. Factor de complejidad técnica.

El factor de complejidad técnica resulta:

$$TCF = 0.6 + 0.01 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$TCF = 0.6 + 0.01 \times (0+4+2+2+5+1.5+1.5+6+3+5+4+0+2)$$

$$TCF = 0.6 + 0.01 \times 36$$

$$TCF = 0.96$$

### Factor de ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del factor de ambiente. El cálculo del mismo es similar al cálculo del factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.



Factor	Descripción	Peso	Valor asignado	Peso * Valor
E1	Familiaridad con el modelo de proyecto utilizado	1.5	0	0
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	4	8
E7	Personal part-time	-1	2	-2
E8	Dificultad del lenguaje de programación	-1	2	-2

Tabla 2.28. Factor de ambiente.

El factor ambiente resulta:

$$EF = 1.4 - 0.03 \times \sum (\text{Peso} \times \text{Valor asignado})$$

$$EF = 1.4 - 0.03 \times (0+1.5+4+2.5+5+8-3-2)$$

$$EF = 1.4 - 0.03 \times 17$$

$$EF = 0.51$$

Finalmente el resultado de los puntos por caso de uso es:

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 101 \times 0.96 \times 0.51$$

$$UCP = 49.4496$$

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

Donde:

**E:** esfuerzo estimado en horas-hombre

**UCP:** Puntos de Casos de Uso ajustados

**CF:** factor de conversión

$$E = 49.4496 \times 20$$

$$E = 998.992$$

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso.

Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

Factor	Descripción	Horas - Hombre
Análisis	10%	249.748
Diseño	20%	499.496
Implementación	40%	998.992
Prueba	15%	374.622
Sobrecarga (otras actividades)	15%	374.622
Total	100%	2497.48

Tabla 2.29. Distribución del Esfuerzo.

Esfuerzo Total de Horas –Hombre: 2497.48

Para todo el desarrollo del trabajo se necesita un total de 2497.48 horas-hombres, por lo tanto considerando que el equipo de trabajo consta de 2 personas y se estima que un mes tiene 4 semanas y cada semana 6 días laborales, por tanto se trabajarían 24 días al mes, si se trabajan 7 horas diarias, como promedio en un mes se trabajarían 168 horas, eso daría como resultado:

$$ET = 14.86 \text{ mes} - \text{hombre.}$$

Si:

$$\text{Tiempo} = ET/CH$$

$$\text{Tiempo} = 14.86/2$$

$$\text{Tiempo} = 7.43$$

Se llega a la conclusión de que con 2 hombres trabajando en el desarrollo del sistema, el tiempo de duración sería de aproximadamente 7 meses y medio.

### 2.13. Conclusiones

En este capítulo se realizaron los procesos del negocio, se identificaron los requerimientos funcionales y no funcionales del sistema, lo que dio lugar a una correcta identificación de los casos de uso del sistema necesarios para satisfacer los requerimientos de la propuesta de solución presentada. Con todo este proceso se muestra con mayor claridad cómo analizar e implementar dicho sistema. Además se realizó la estimación del esfuerzo en todo el ciclo de trabajo obteniéndose resultados factibles en un corto período de tiempo.

## CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA

### 1.1. Introducción

El siguiente capítulo se encuentra dividido en dos partes fundamentales: en la primera se realiza el análisis del sistema donde se realizan los casos de uso teniendo en cuenta los requisitos funcionales que se especificaron en el capítulo anterior.

En la segunda parte se procede al diseño de la aplicación teniendo en cuenta el patrón de arquitectura a utilizar, de modo que se realicen los casos de uso incluyendo los requisitos no funcionales antes definidos.

### 1.2. Diagramas de clases de análisis

Durante el análisis, analizamos los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema.

Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Los estereotipos de las mismas son los siguientes:

Clase interfaz: Modelan información que posee larga vida y que es a menudo persistente.

Clase controladora: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

Clase entidad: Modelan la interacción entre el sistema y sus actores.

El Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. (10)

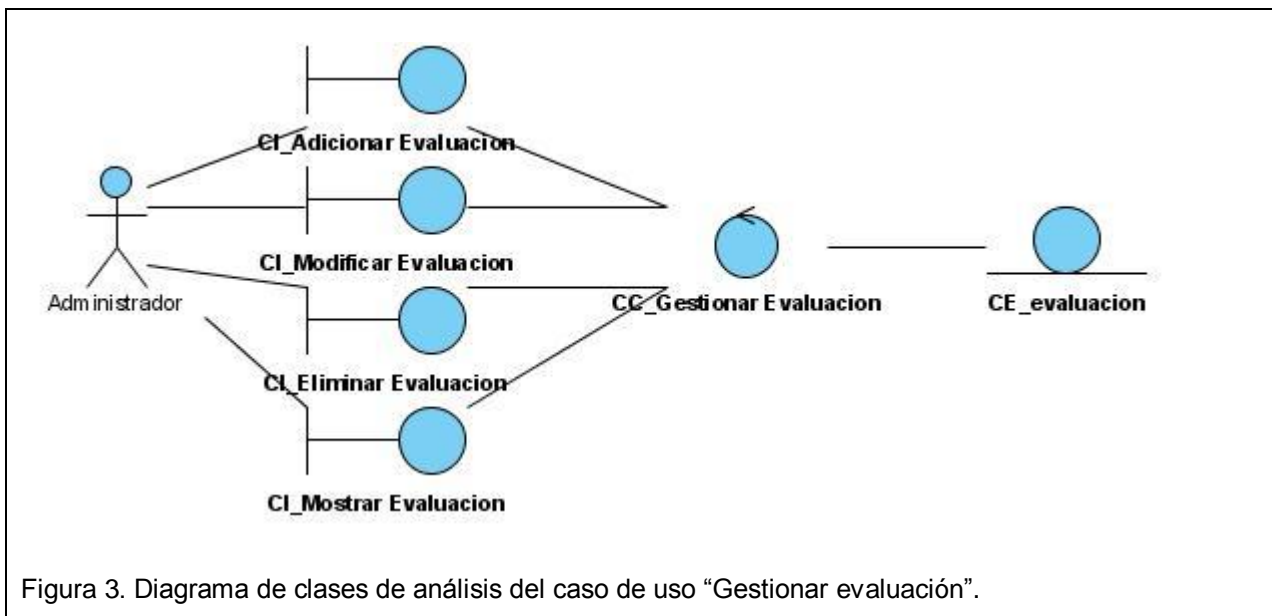


Figura 3. Diagrama de clases de análisis del caso de uso “Gestionar evaluación”.

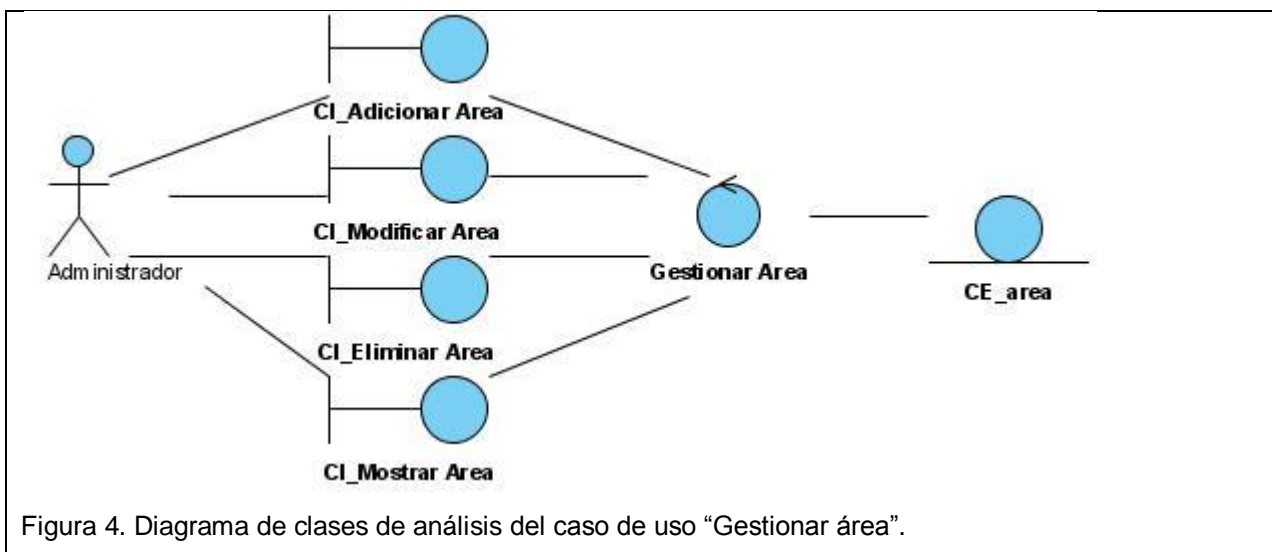


Figura 4. Diagrama de clases de análisis del caso de uso “Gestionar área”.

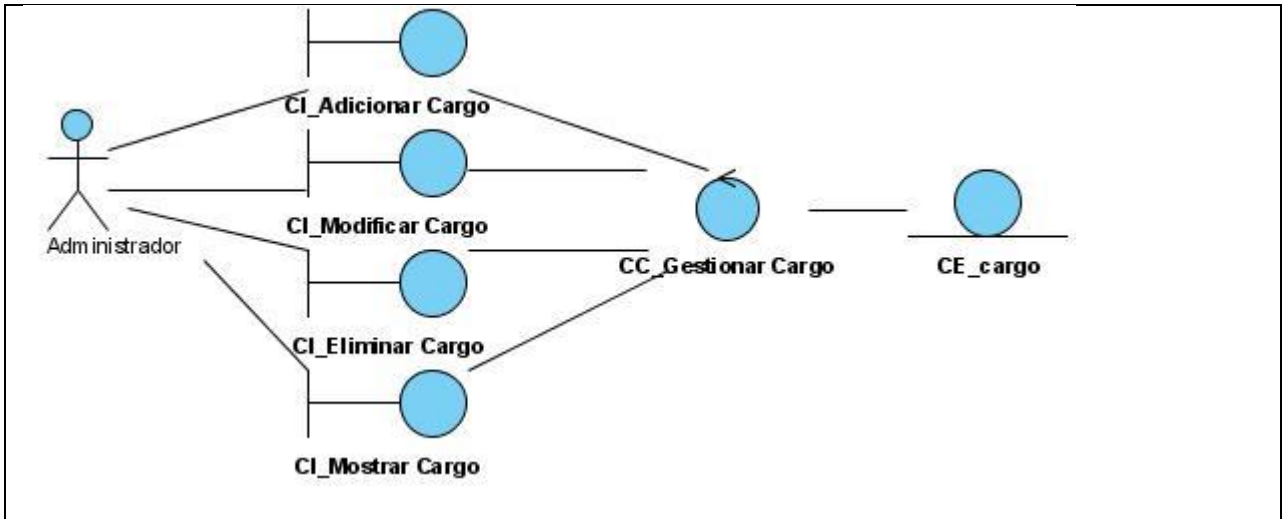


Figura 5. Diagrama de clases del análisis del caso de uso “Gestionar cargo”.

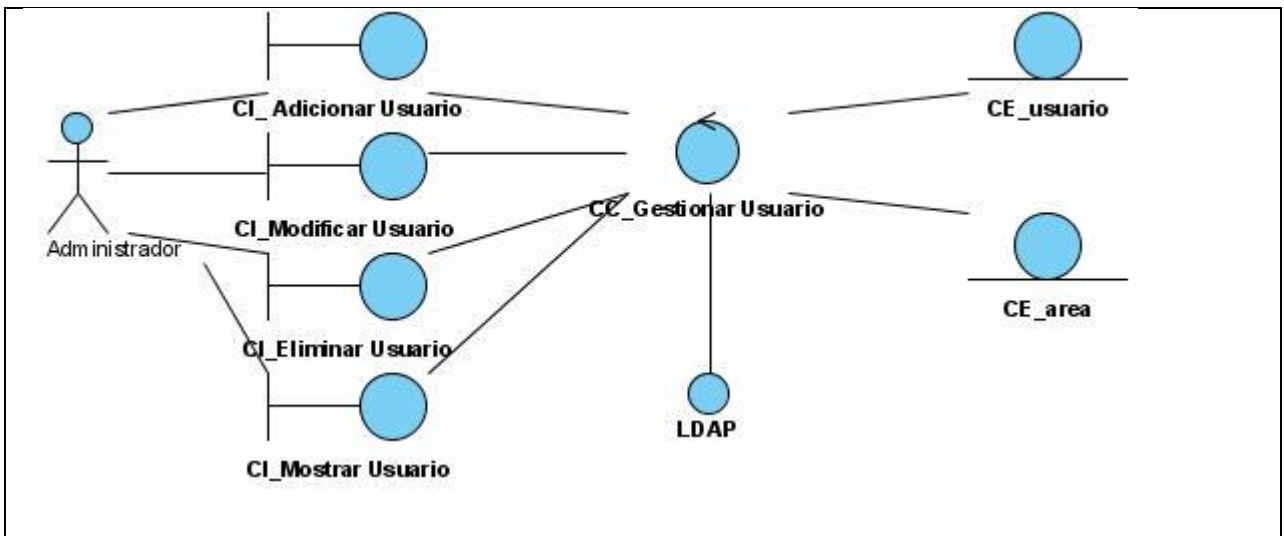


Figura 6. Diagrama de clases del análisis del caso de uso “Gestionar usuario”.

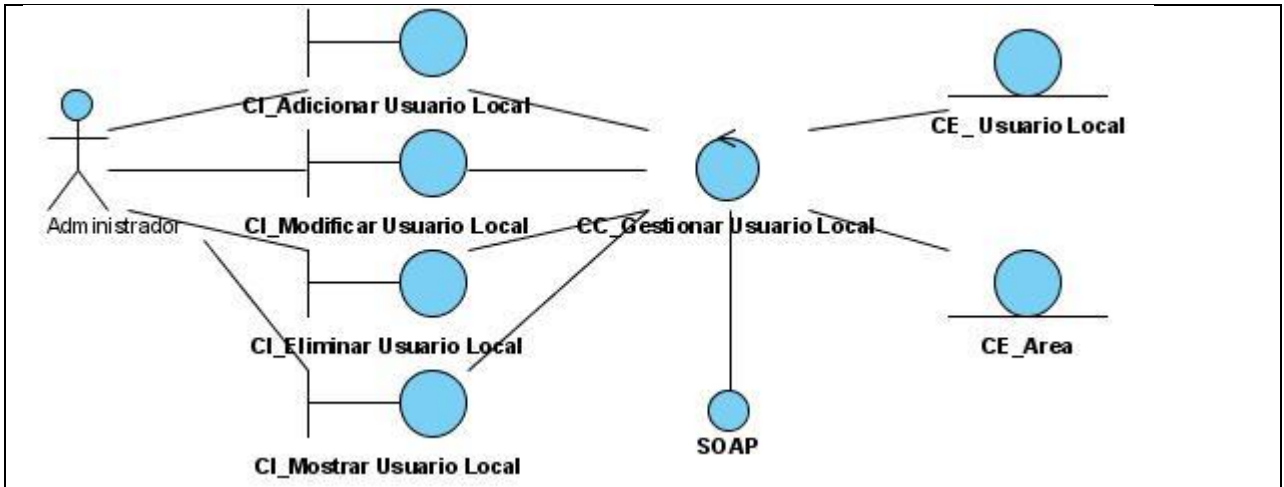


Figura 7. Diagrama de clases del análisis del caso de uso “Gestionar usuario local”.

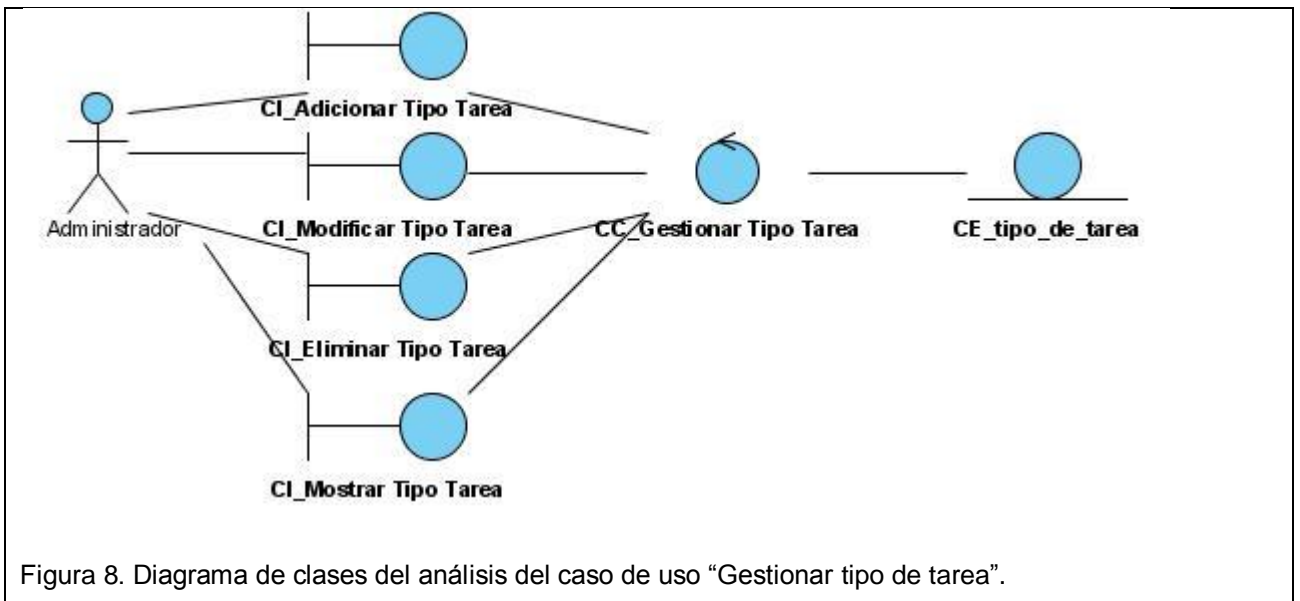
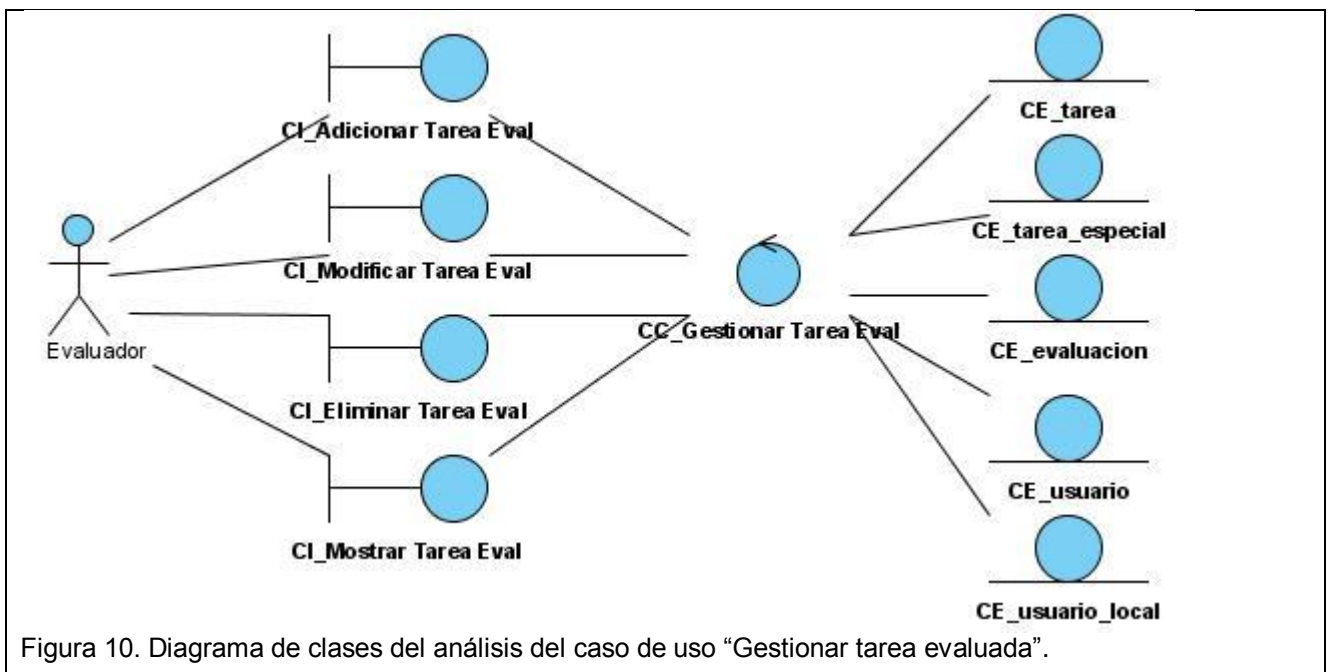
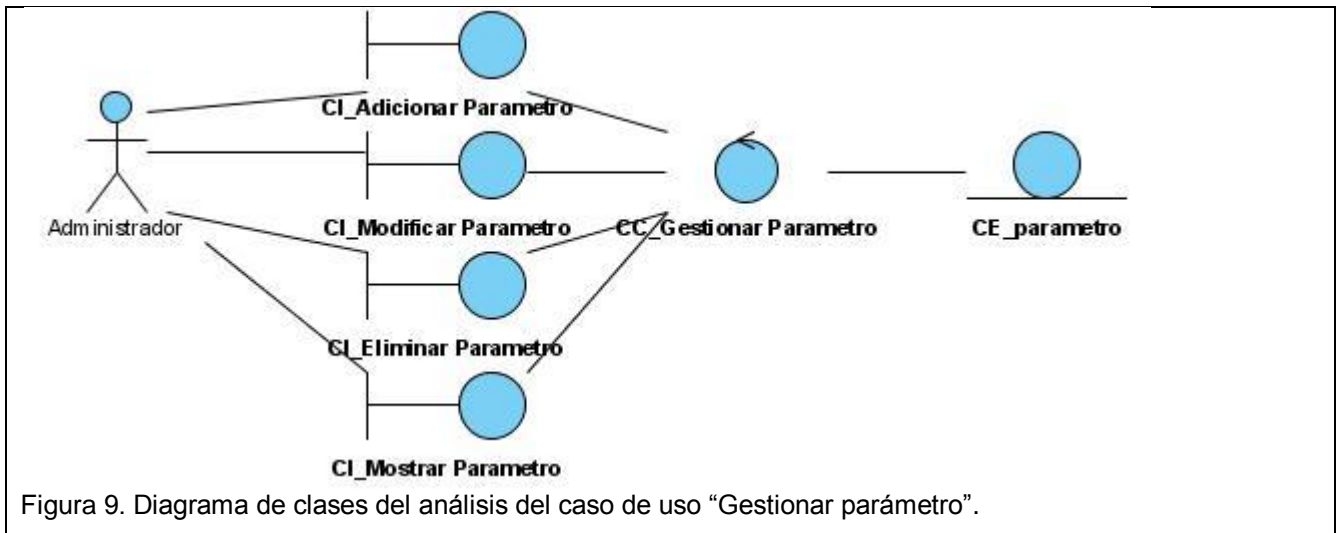


Figura 8. Diagrama de clases del análisis del caso de uso “Gestionar tipo de tarea”.





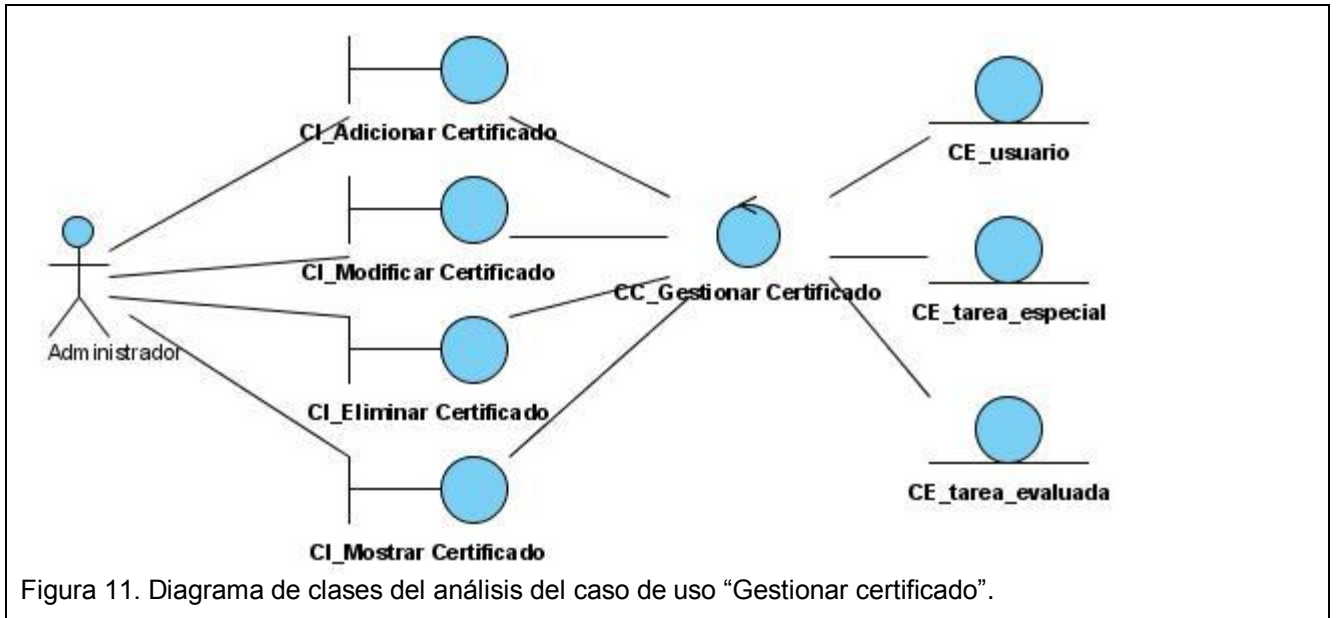


Figura 11. Diagrama de clases del análisis del caso de uso "Gestionar certificado".

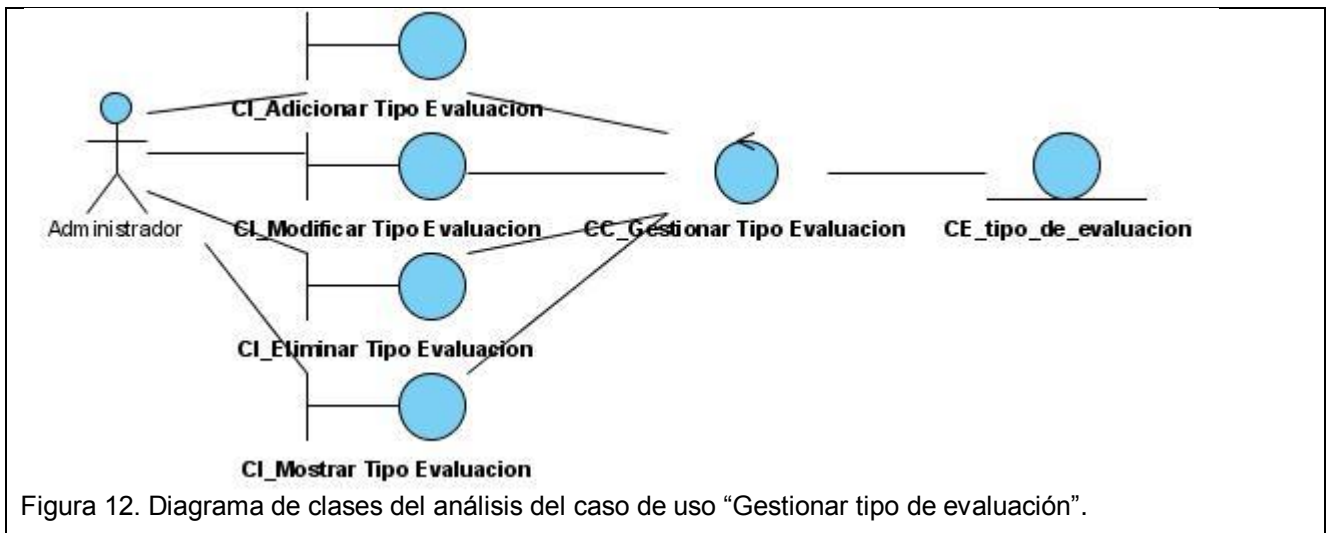
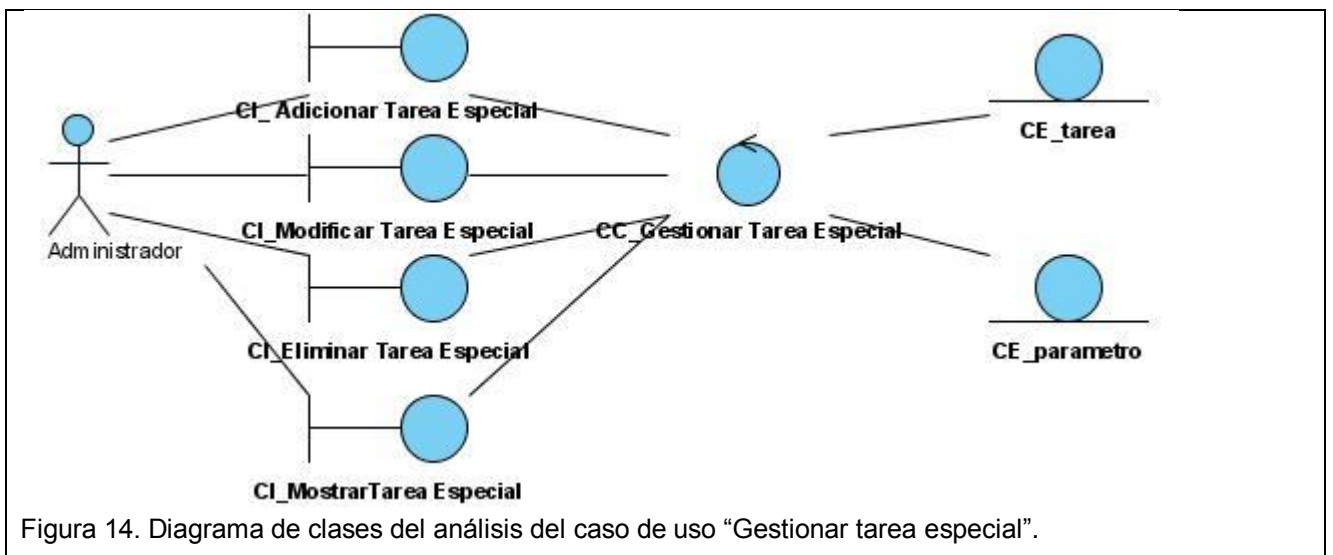
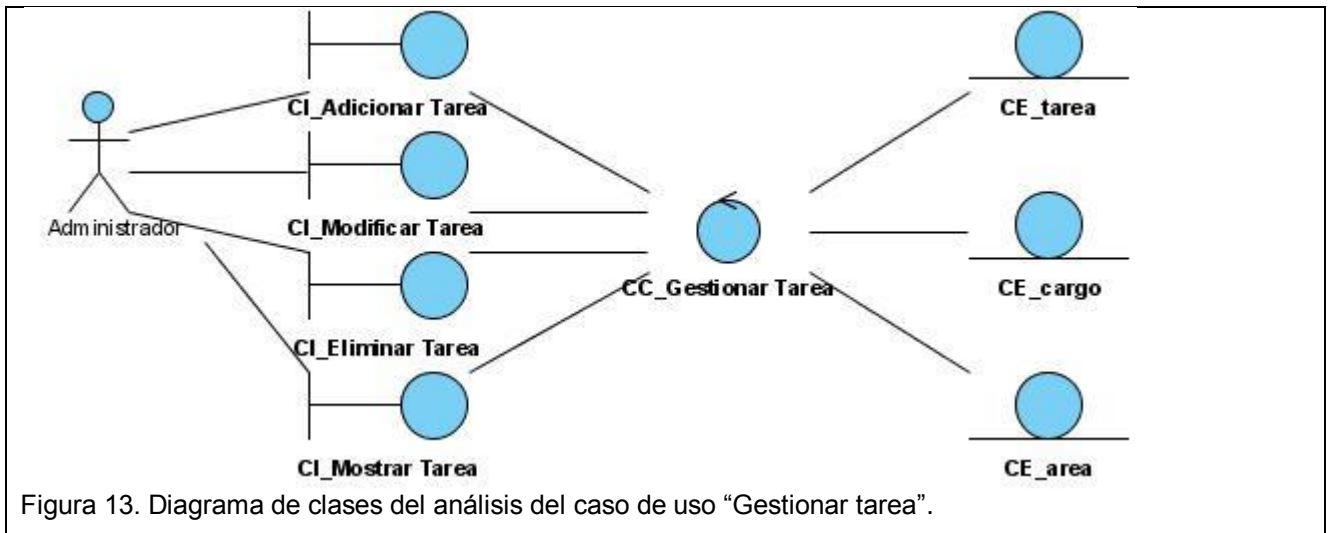
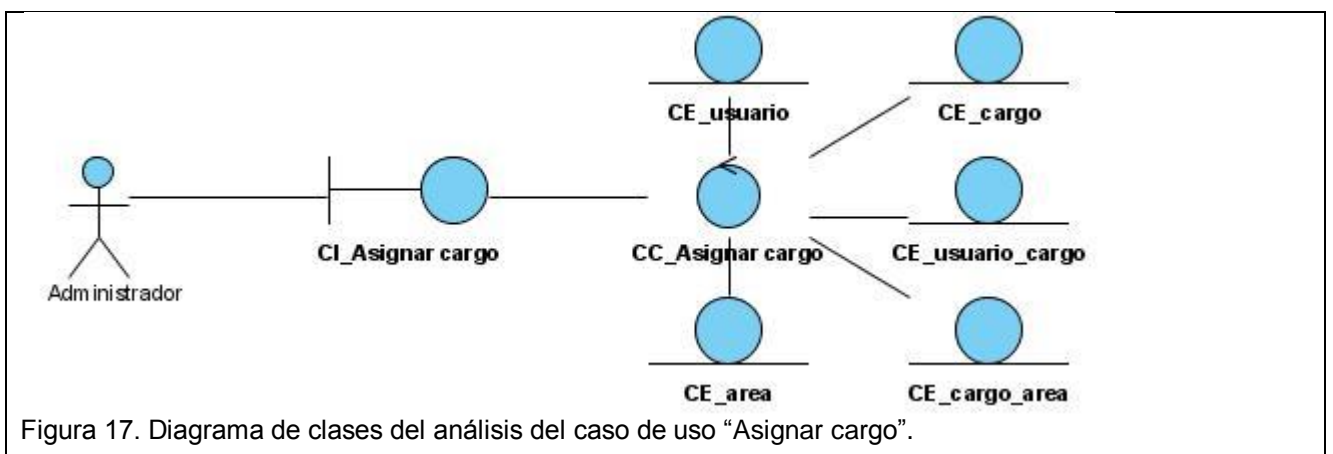
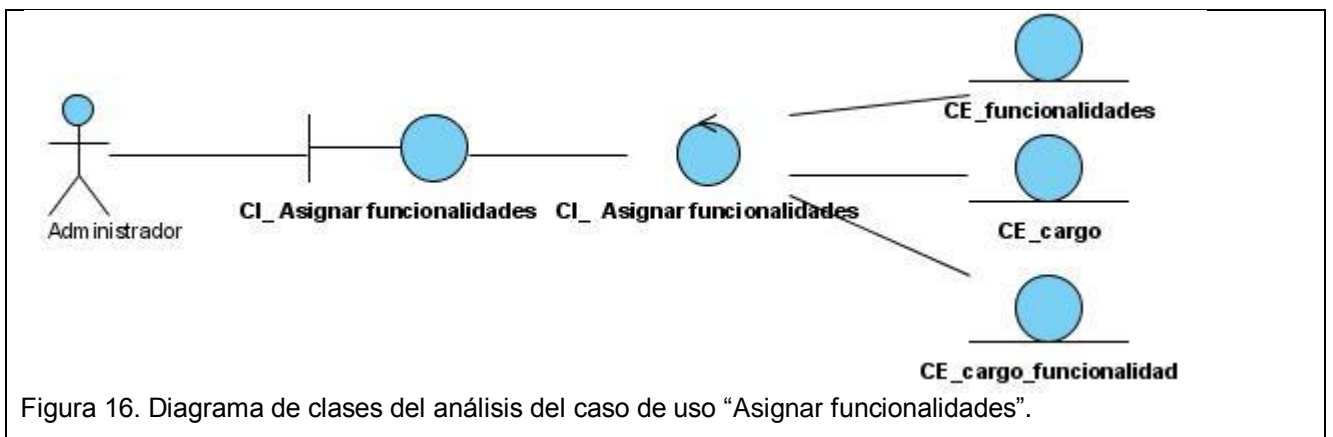
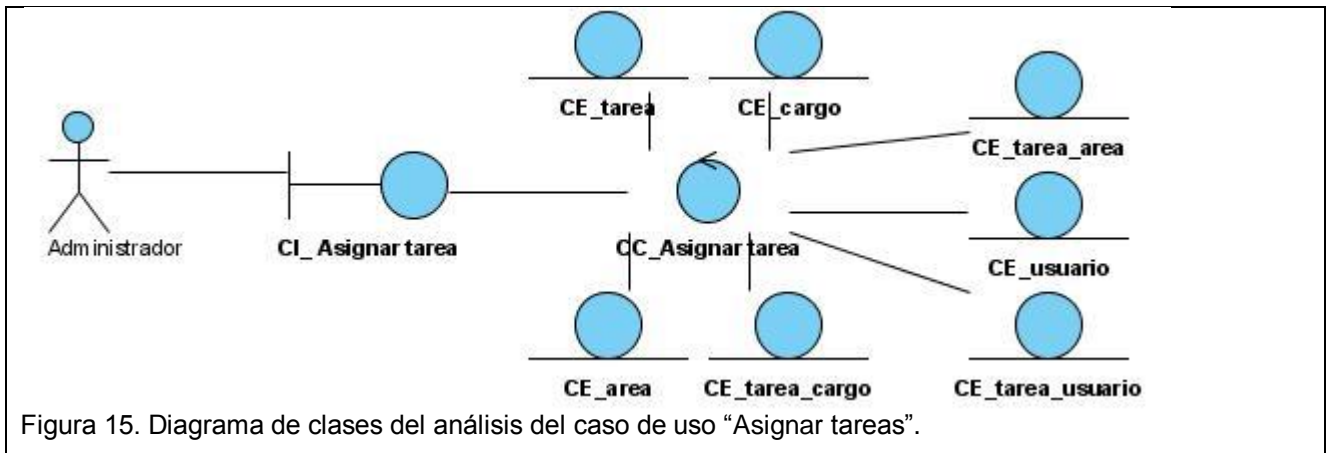


Figura 12. Diagrama de clases del análisis del caso de uso "Gestionar tipo de evaluación".





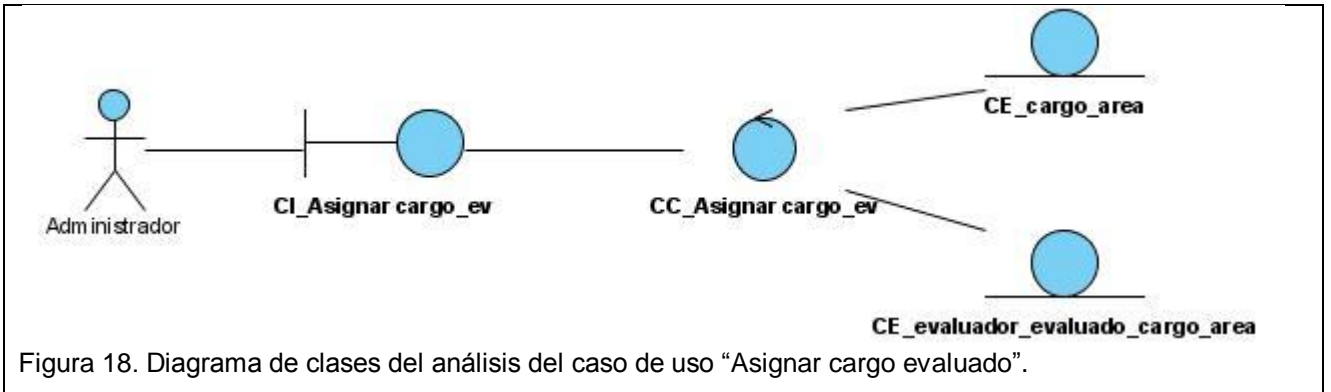


Figura 18. Diagrama de clases del análisis del caso de uso "Asignar cargo evaluado".

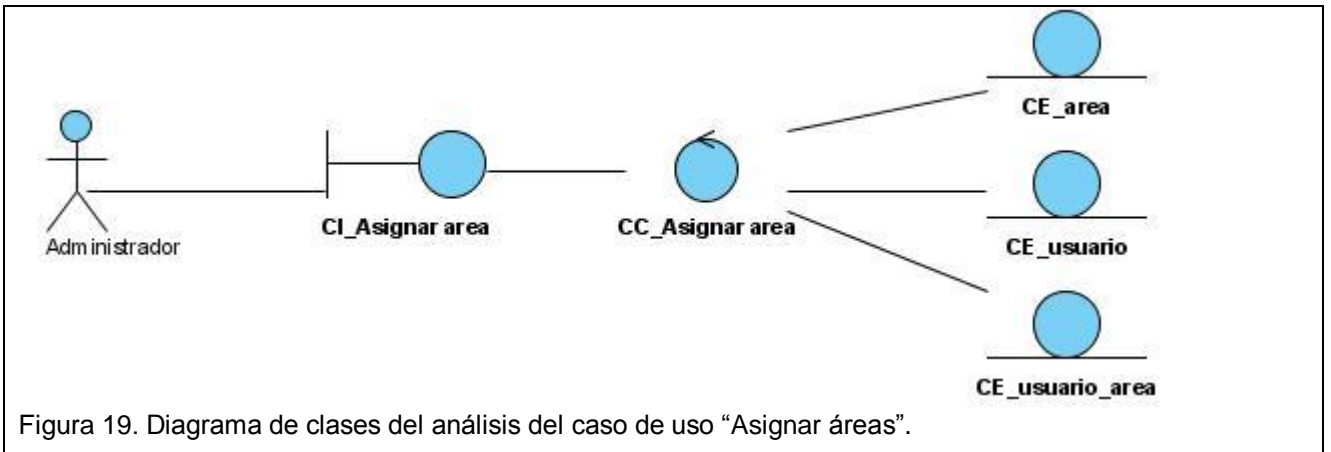


Figura 19. Diagrama de clases del análisis del caso de uso "Asignar áreas".

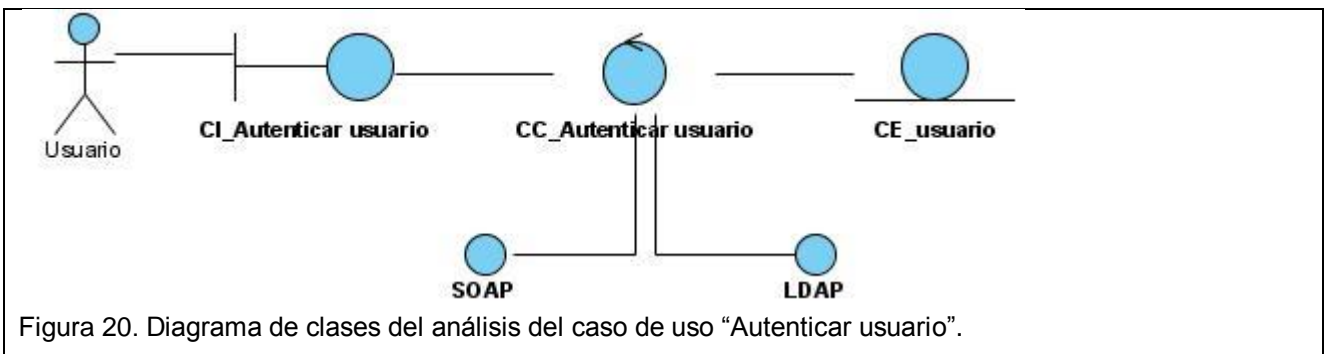


Figura 20. Diagrama de clases del análisis del caso de uso "Autenticar usuario".

### 1.3. Aplicación de algunos patrones GRASP en Symfony

#### Patrón Experto

En la arquitectura de symfony, en el modelo, existen 2 tipos de clases, las encargadas de la abstracción de la base de datos y las de acceso a datos. Symfony genera 4 clases por cada tabla de la base de datos, por ejemplo si se tiene una tabla llamada tb\_usuario, se obtendrían las clases: Tusuario, BaseTusuario, TusuarioPeer y BaseTusuarioPeer. Las clases de abstracción de los datos serían las de tipo "Peer", que son las que trabajan directamente con la base de datos. Son las que tienen los atributos necesarios para obtener los registros de las tablas de la base de datos, en esa interacción directa es donde se pone de manifiesto el patrón experto.

#### Patrón Alta Cohesión

El utilizar Symfony, permite crear y trabajar con clases con una alta cohesión ya que posibilita organizar el trabajo de acuerdo a la estructura del proyecto que se esté realizando. Un ejemplo claro es la clase actions.class.php, que son las responsables de controlar las acciones de las plantillas.

#### Patrón Bajo Acoplamiento

Debe haber pocas dependencias entre las clases para poder extraerlas de un modo independiente y reutilizarlas. Este patrón se pone de manifiesto a partir de que el modelo posee las clases que implementan la lógica de negocio y acceso a datos, y estas clases no tienen relaciones con las de la vista o el controlador por lo que la dependencia es baja.

#### Patrón Controlador

Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Este patrón se pone de manifiesto en las clases sfFrontController, sfWebFrontController, sfContext, los "actions" y el index.php del ambiente. La arquitectura del marco de trabajo (MVC) brinda una capa específicamente para los controladores, que son el núcleo del mismo y especifica la presencia de este patrón.

### 1.4. Descripción de las clases del diseño

<b>Nombre de la Clase:</b> myUser.class.php
<b>Nombre del método:</b> DetectarRol(\$id_usuario)
<b>Descripción:</b> Identifica si el usuario logueado, es administrador, evaluador y evaluado, evaluador, evaluado o invitado.
<b>Nombre del método:</b> ObtenerUsuarioLocal(\$user,\$pass)
<b>Descripción:</b> Verifica si el usuario tiene una cuenta local.
<b>Nombre del método:</b> ObtenerUsuarioDelDominio(\$solapin)
<b>Descripción:</b> Verifica en caso de que el usuario se haya logueado correctamente, con usuario y contraseña del dominio, si su solapín existe en la base de datos. Devuelve el objeto usuario.
<b>Nombre del método:</b> ObtenerFuncionalidades()
<b>Descripción:</b> obtiene dado cada cargo del usuario logueado, sus funcionalidades.
<b>Nombre del método:</b> ObtenerCargoReal(\$cargos)
<b>Descripción:</b> Obtiene el cargo real de un usuario.
<b>Nombre del método:</b> ObtenerCargos(\$solapin)
<b>Descripción:</b> Obtiene todos los cargos de un usuario que se haya logueado dado su solapín.
<b>Nombre del método:</b> ObtenerModulos()
<b>Descripción:</b> Obtiene el módulo al que pertenece cada funcionalidad.
<b>Nombre del método:</b> Logging (\$user,\$pass)
<b>Descripción:</b> Verifica la veracidad del usuario y la contraseña que se insertan a la hora de loguearse.

Tabla 2.30. Descripción de la clase myUser.

<b>Nombre del módulo:</b> autenticar
<b>Nombre de la Clase:</b> actions.class.php
<b>Nombre del método:</b> executeLogout()
<b>Descripción:</b> Desautentica al usuario.
<b>Nombre del método:</b> executeIndex()
<b>Descripción:</b> Redirecciona a executeEntra_datos().
<b>Nombre del método:</b> executeEntra_datos()
<b>Descripción:</b> Llama a la plantilla que contiene el formulario para autenticarse.
<b>Nombre del método:</b> executeMuestra_datos()
<b>Descripción:</b> Recibe los datos de autenticación, si están correctos, muestra la página de inicio según el usuario que se haya autenticado, sino ordena que se muestre un mensaje de error en la plantilla.

Tabla 2.31. Descripción de la clase actions del módulo de autenticación.

<b>Nombre del módulo:</b> informacion
<b>Nombre de la Clase:</b> actions.class.php
<b>Nombre del método:</b> executeIndex()
<b>Descripción:</b> Le da acceso al indexSuccess a los datos informativos que se le quieran mostrar al usuario.
<b>Nombre del método:</b> executeGuardar()
<b>Descripción:</b> Recibe la información que ha sido editada y la guarda, para mostrarla en el indexSuccess.
<b>Nombre del método:</b> executeArchivo()
<b>Descripción:</b> Permite subir un archivo para que sea descargado o consultado por el usuario.
<b>Nombre del método:</b> executePdf()
<b>Descripción:</b> Permite generar un documento en formato pdf la planilla de evaluación por desempeño de un usuario evaluado.

Tabla 2.32. Descripción de la clase actions del módulo de información.



<b>Nombre del módulo:</b> error
<b>Nombre de la Clase:</b> actions.class.php
<b>Nombre del método:</b> executePaginaNoEncontrada()
<b>Descripción:</b> Llama a la página paginaNoEncontradaSuccess y muestra un mensaje diciendo que la página a la que se desea acceder no fue encontrada.
<b>Nombre del método:</b> executeNoCredenciales()
<b>Descripción:</b> Llama a la página noCredencialesSuccess y muestra un mensaje de insuficiencia de privilegios para ejecutar una operación dada.

Tabla 2.33. Descripción de la clase actions del módulo de errores.

<b>Nombre de los módulos:</b> gest_area gest_cargo gest_certificado gest_evaluacion gest_parametro gest_tarea gest_tarea_especial gest_tarea_evaluada gest_tipo_de_evaluacion gest_tipo_de_tarea gest_usuario gest_usuario_local
<b>Nombre de la Clase:</b> actions.class.php
<b>Nombre del método:</b> executeIndex()
<b>Descripción:</b> Redirecciona para el método executeList().
<b>Nombre del método:</b> executeList()
<b>Descripción:</b> Guarda todos los elementos de la tabla en una variable para enviarlas al listSuccess de la vista. Los atributos de la clase de la acción (disponibles vía <code>\$this-&gt;nombreDeVariable</code> en la acción) estén accesibles en la plantilla (vía <code>nombreDeVariable</code> )
<b>Nombre del método:</b> executeCreate()

<b>Descripción:</b> Crea un nuevo objeto de la tabla y permite que sea accesible para el editSuccess de la vista para que sea creado.
<b>Nombre del método:</b> executeEdit()
<b>Descripción:</b> Recibe un id de la tabla, obtiene un objeto dado ese id y permite que sea accesible por el editSuccess de la vista para que sea modificado.
<b>Nombre del método:</b> executeUpdate()
<b>Descripción:</b> Recibe los datos y guarda los datos que han sido creados o editados en el createSuccess o EditSuccess respectivamente.
<b>Nombre del método:</b> executeDelete()
<b>Descripción:</b> Recibe un id y en caso de encontrar un objeto con este id lo elimina.

Tabla 2.34. Descripción de las clases actions de los módulos construidos para gestionar los nomencladores de la base de datos.

<b>Nombre de los módulos:</b> gest_cargo_area gest_cargo_funcionalidad gest_evaluador_evaluado gest_evaluador_evaluado_cargo_area gest_tarea_area gest_tarea_cargo gest_tarea_cargo_area gest_usuario_cargo
<b>Nombre de la Clase:</b> actions.class.php
<b>Nombre del método:</b> executeIndex()
<b>Descripción:</b> Redirecciona para el método executeList().
<b>Nombre del método:</b> executeShow()
<b>Descripción:</b> Muestra los objetos de que se van a asignar a una primera tabla
<b>Nombre del método:</b> executeList()
<b>Descripción:</b> Da acceso a que se pueda mostrar a través del listSuccess, un listado de los objetos de la tabla a la cual se le van a asignar, objetos de una segunda tabla.

<b>Nombre del método:</b> executeUpdate()
<b>Descripción:</b> Elimina las relaciones en la tabla relacional y crea nuevas a partir de los datos recibidos.

Tabla 2.35. Descripción de las clases actions de los módulos que gestionan las asignaciones en el sistema.

### 1.5. Diagramas de clases del diseño

En el diseño se modela el sistema de forma tal que le de soporte tanto a los requerimientos funcionales como a los no funcionales. Este modelo se utiliza para visualizar lo que se hará en la implementación, de este modo se brinda una idea más clara de cómo quedaría el sistema propuesto. A continuación se muestran los diagramas de clases del diseño de una muestra de casos de uso que se decidió plasmar en el documento, el resto se pueden encontrar en el Anexo 4.

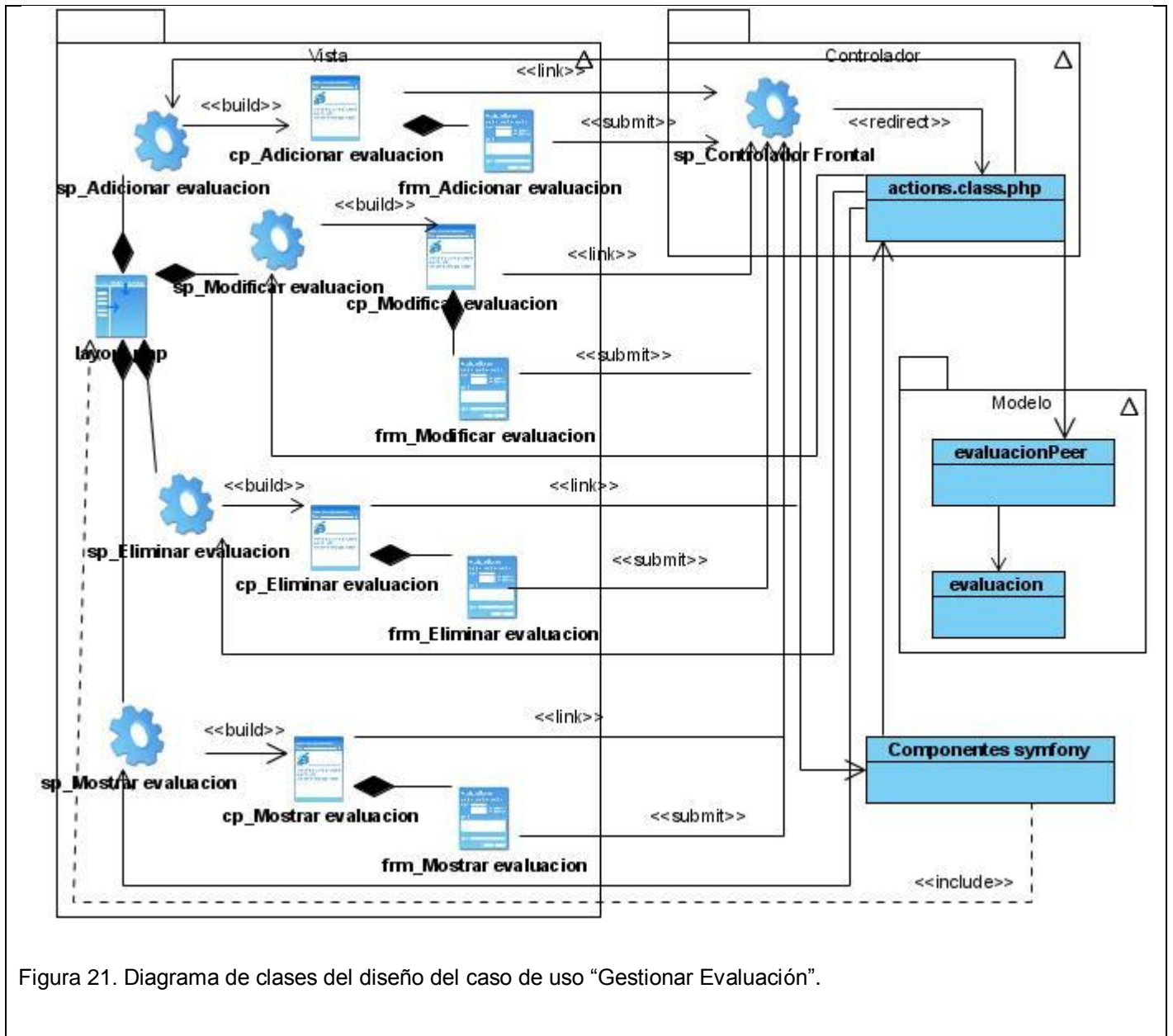


Figura 21. Diagrama de clases del diseño del caso de uso "Gestionar Evaluación".

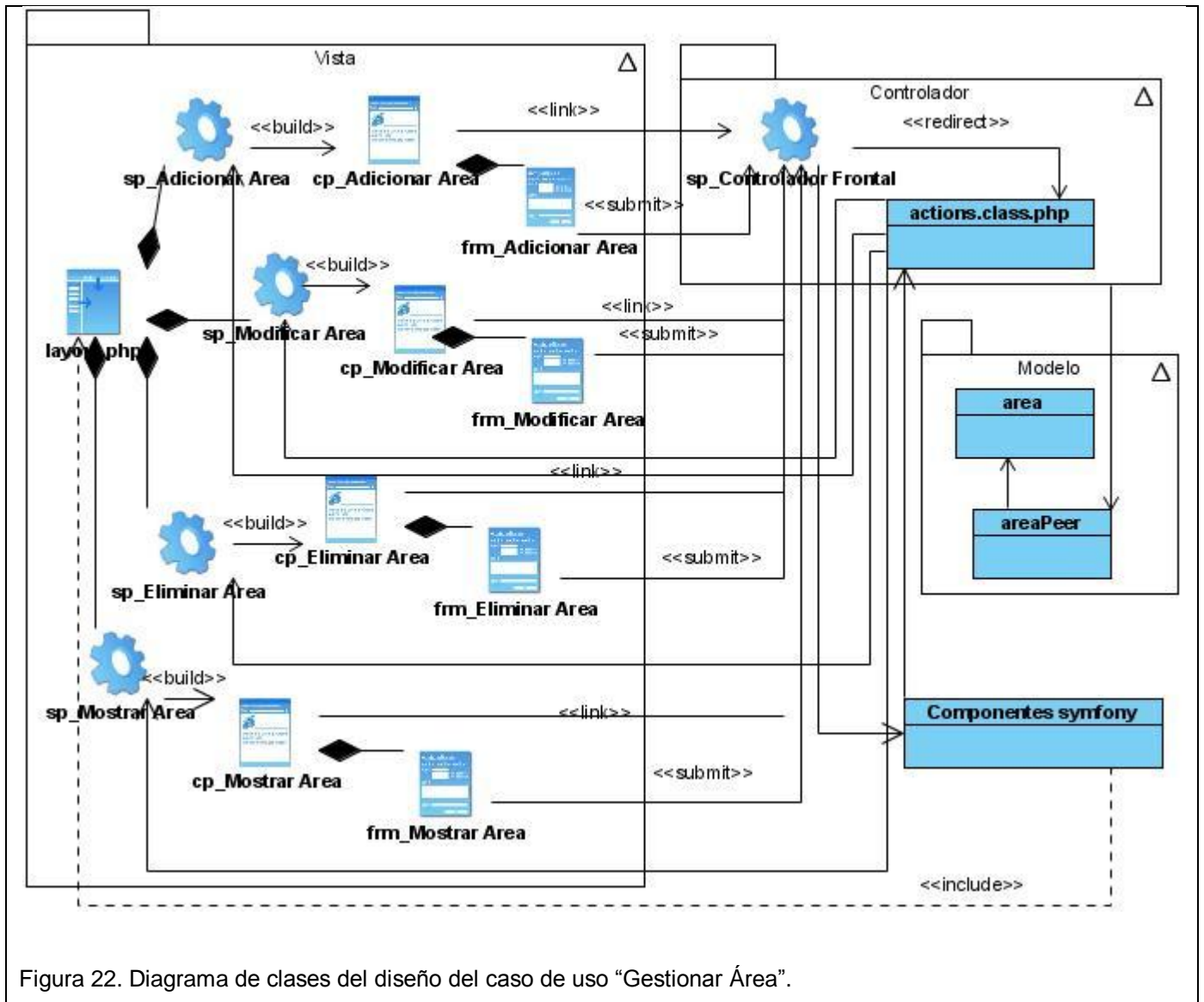


Figura 22. Diagrama de clases del diseño del caso de uso "Gestionar Área".

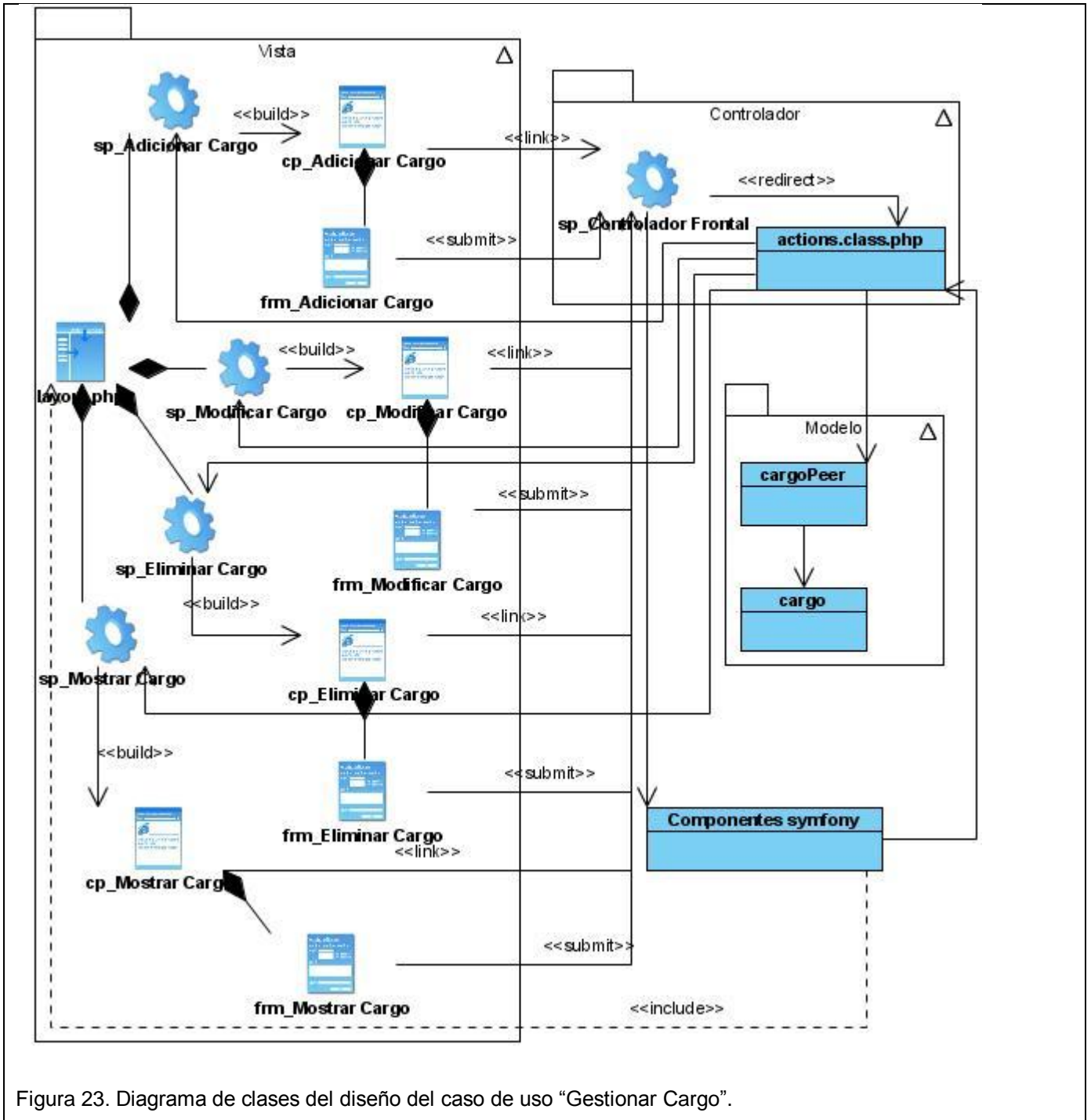


Figura 23. Diagrama de clases del diseño del caso de uso "Gestionar Cargo".

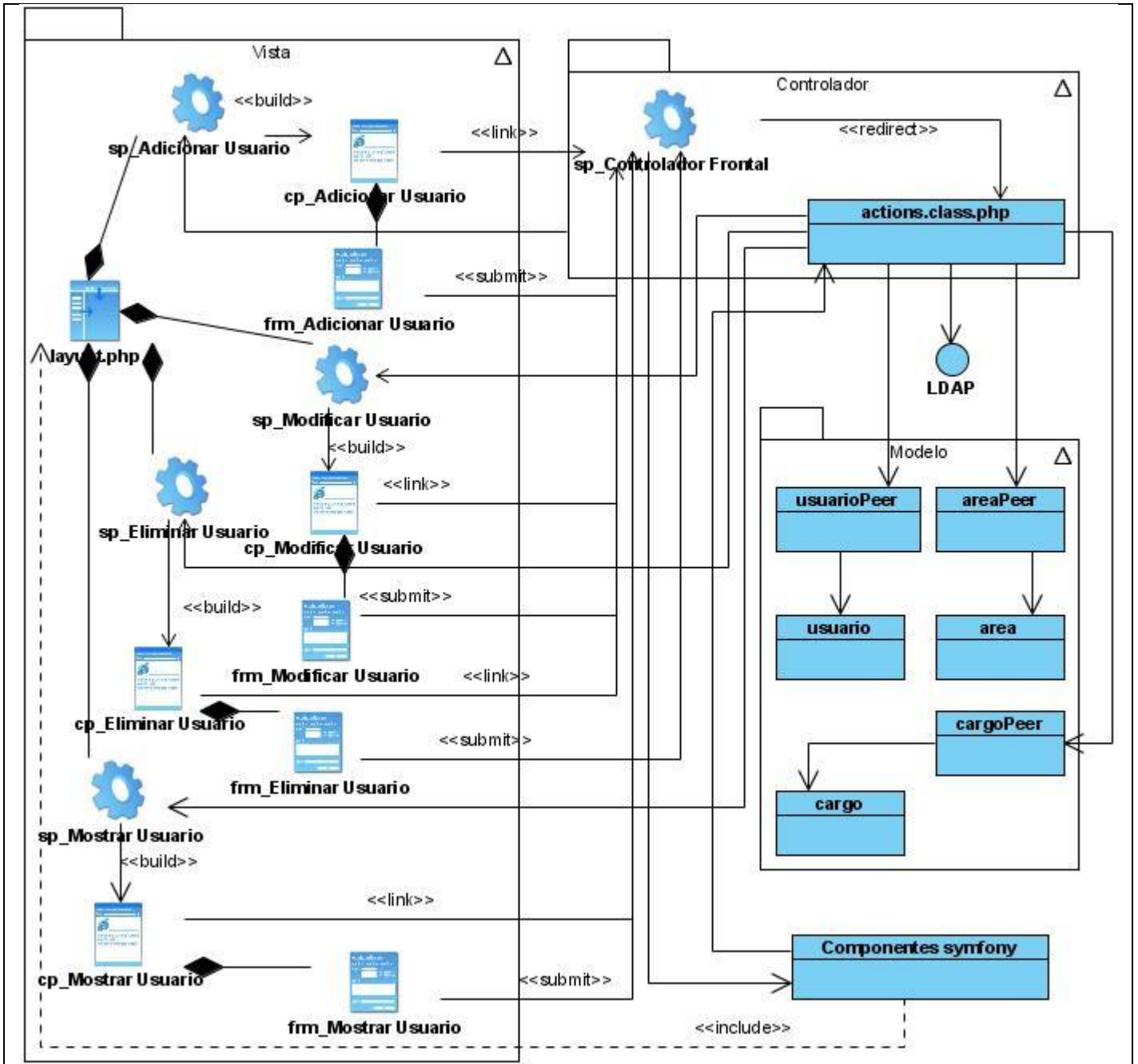


Figura 24. Diagrama de clases del diseño del caso de uso “Gestionar Usuario”.

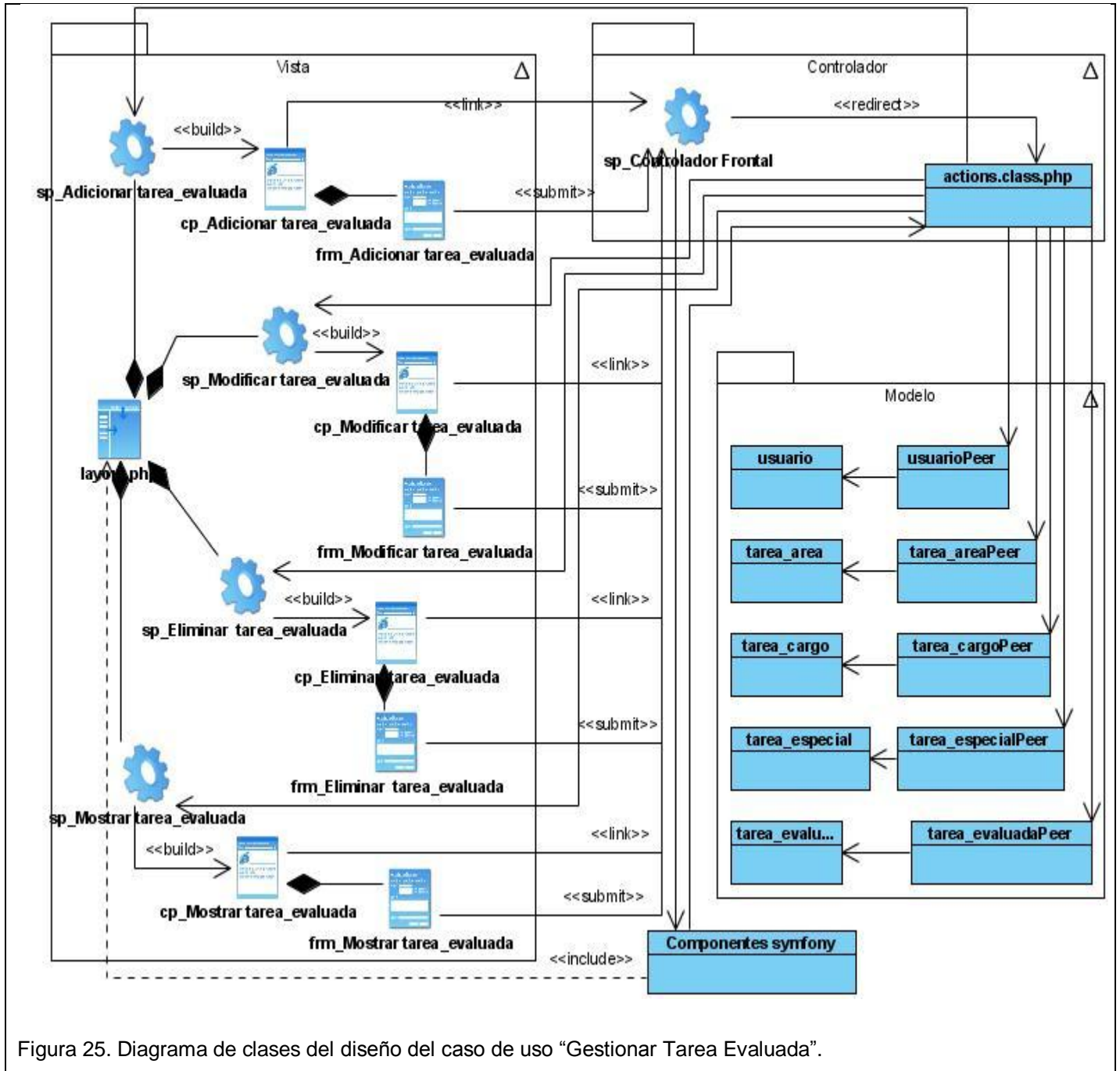


Figura 25. Diagrama de clases del diseño del caso de uso "Gestionar Tarea Evaluada".



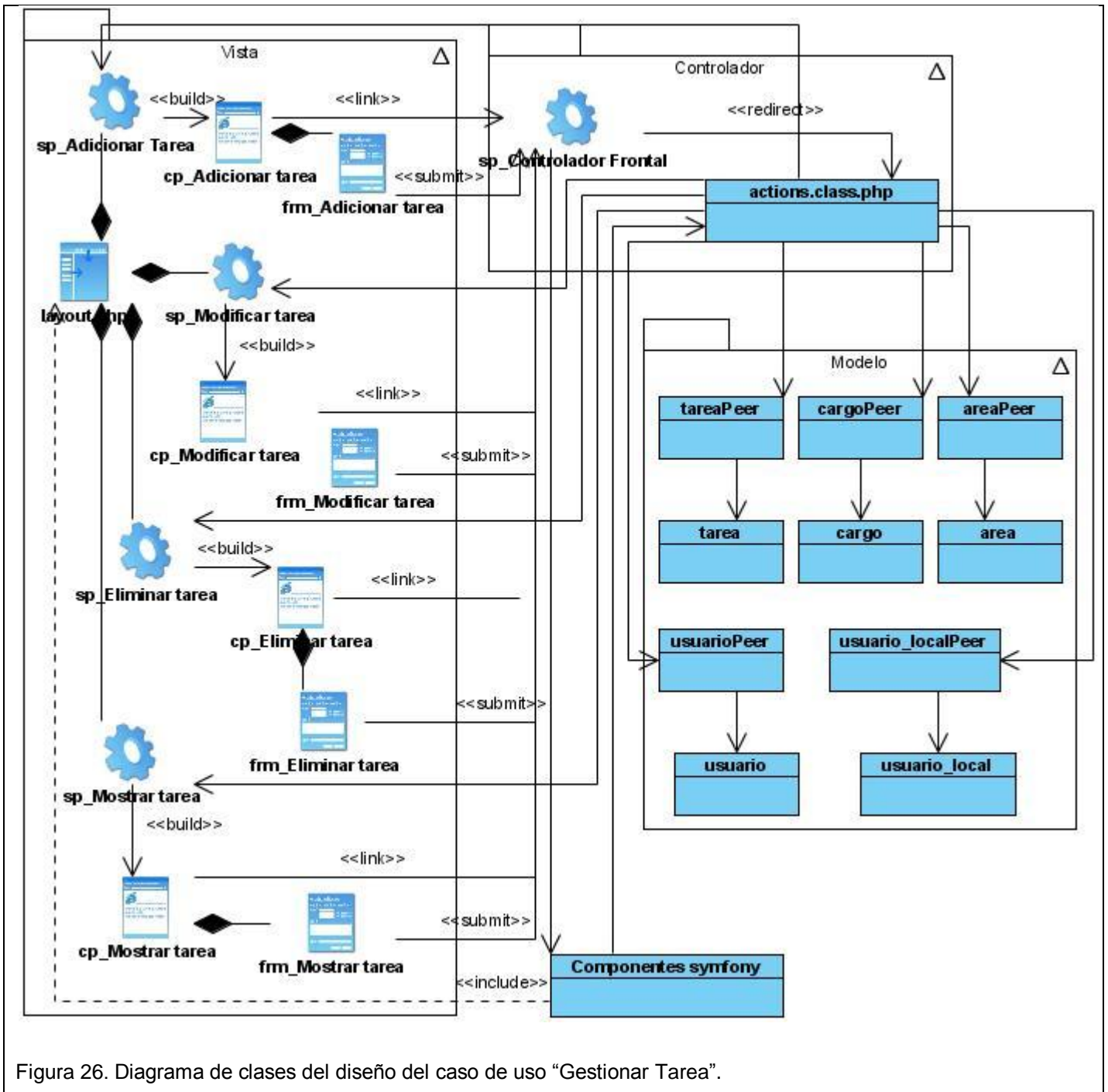


Figura 26. Diagrama de clases del diseño del caso de uso "Gestionar Tarea".

### **1.6. Diseño de la base de datos**

A la hora de realizar el diseño de la base de datos se debe tener en cuenta si se satisfacen los requisitos funcionales definidos en el sistema y se debe velar porque se logre mantener la integridad de los datos y si se brinda la seguridad necesaria. Para diseñar la base de datos se utilizan los modelos lógico y físico de datos. Las clases persistentes que a continuación se muestran cubren las necesidades que el sistema pueda presentar.

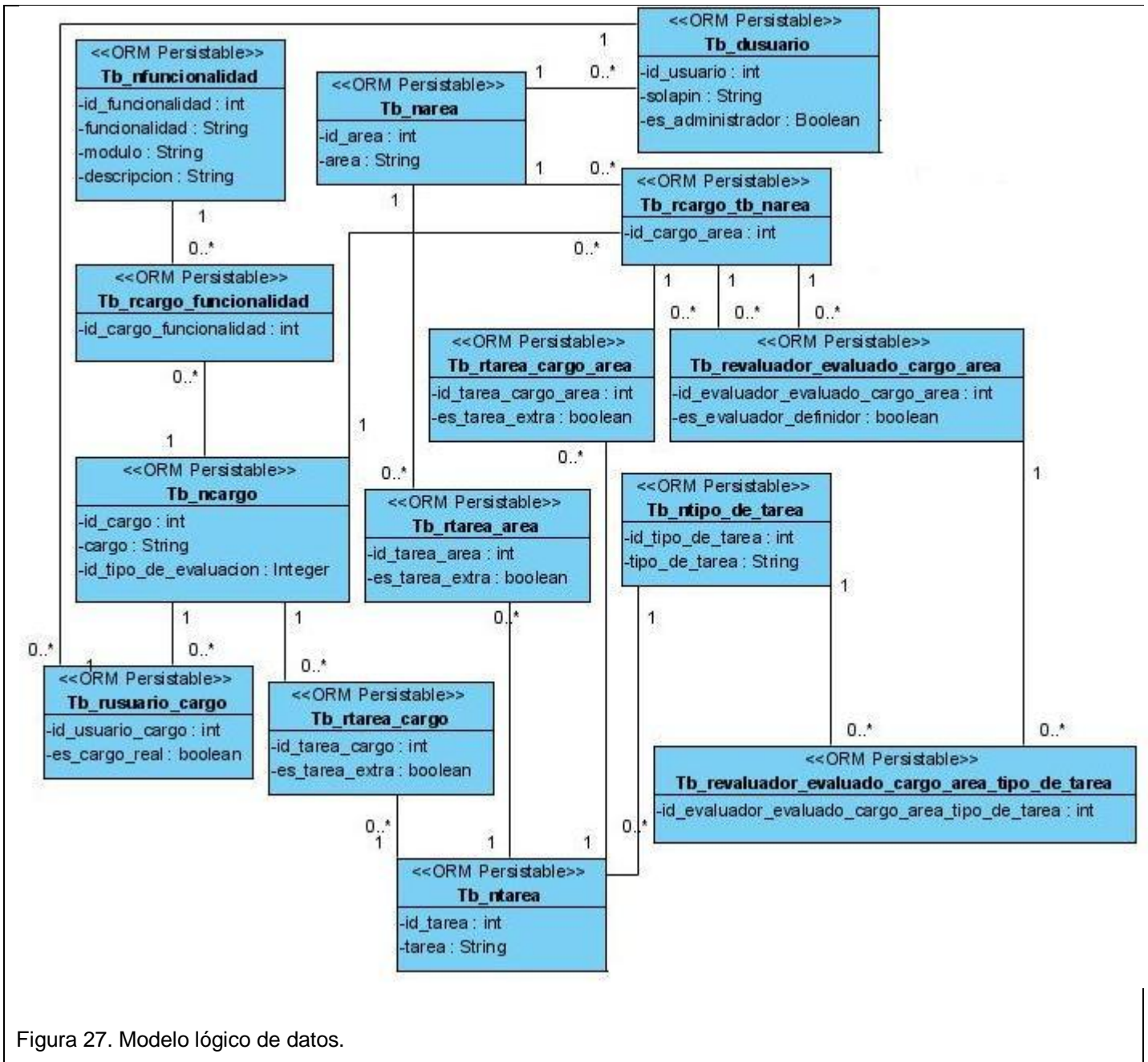


Figura 27. Modelo lógico de datos.

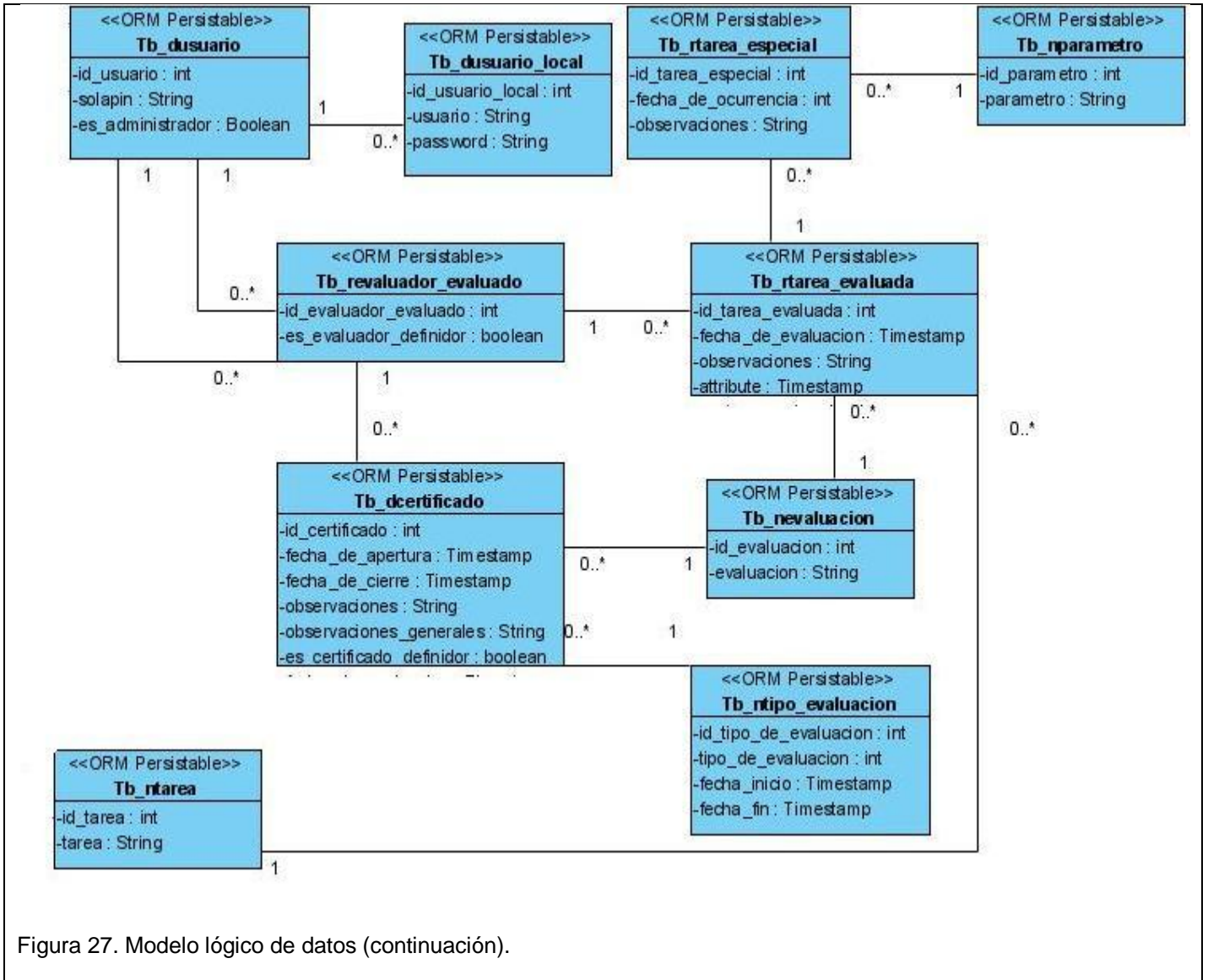


Figura 27. Modelo lógico de datos (continuación).



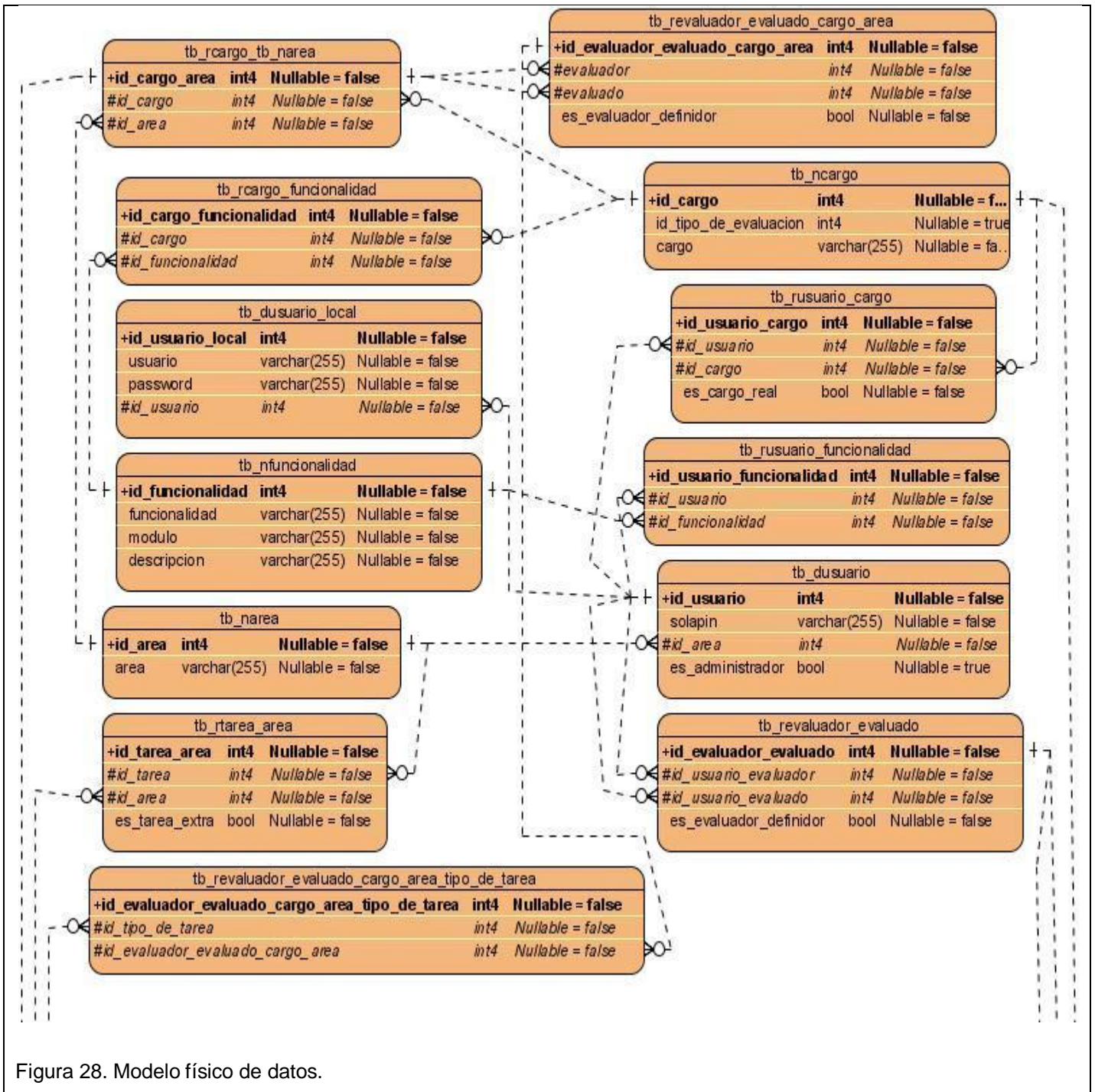


Figura 28. Modelo físico de datos.

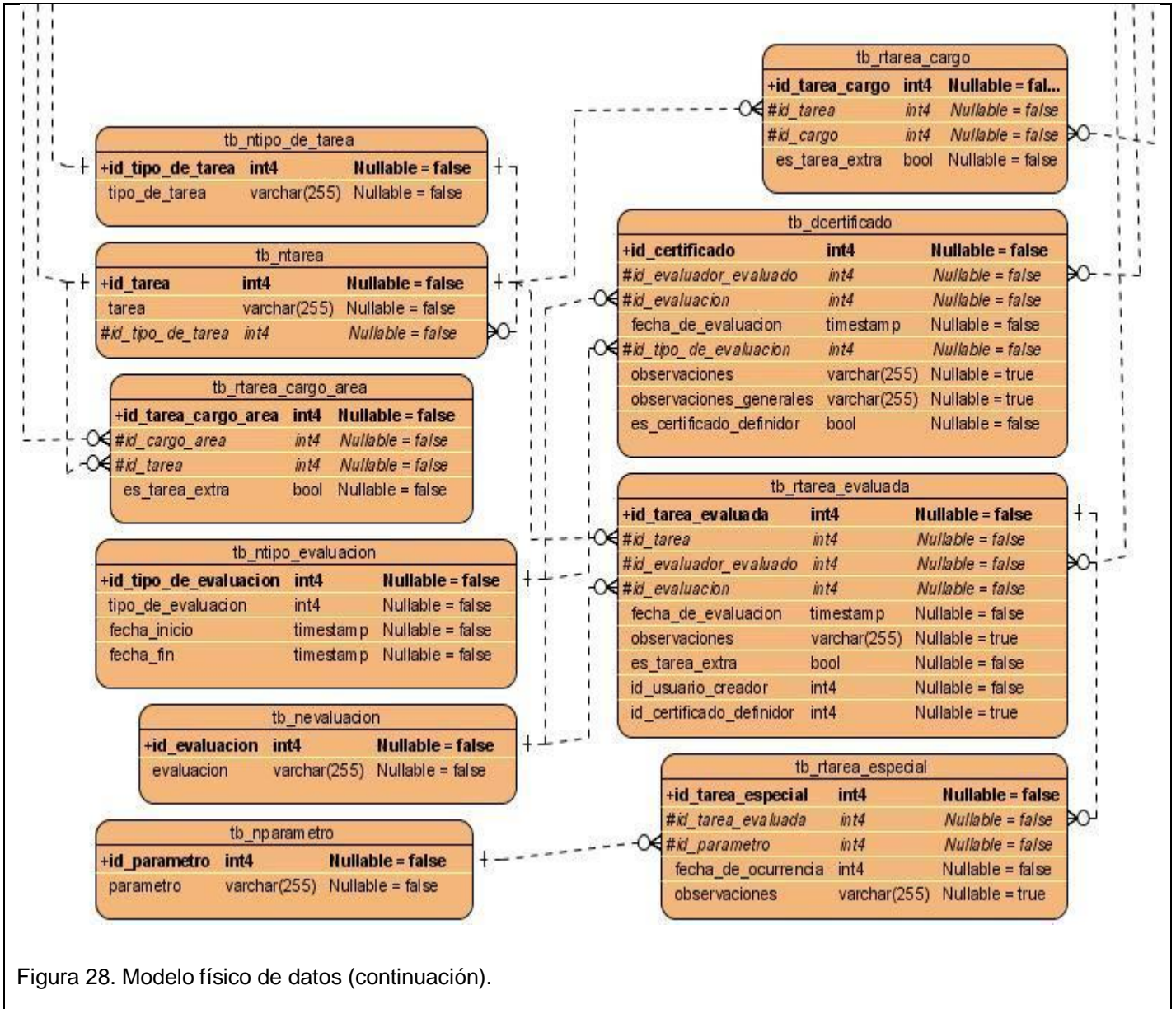
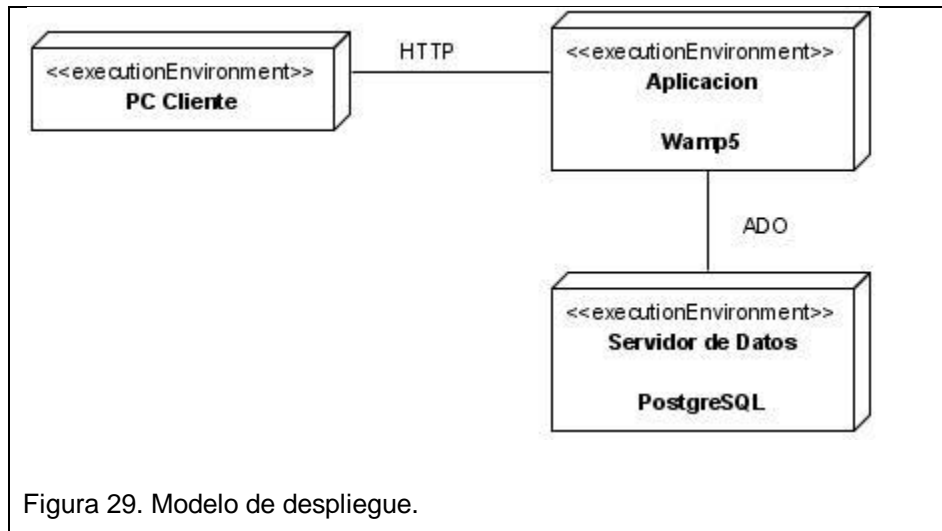


Figura 28. Modelo físico de datos (continuación).

### 1.7. Modelo de despliegue

El modelo de despliegue es un diagrama de objetos que especifica la descripción física del sistema. En el modelo de despliegue que se presenta se define un servidor de base de datos, un servidor de aplicación y un dispositivo para la impresión que va conectado al equipo de cómputo del cliente con el único objetivo de satisfacer sus necesidades.



### 1.8. Tratamiento de errores

En el desarrollo del trabajo cotidiano, los usuarios de nuestro sistema, pueden cometer errores a la hora de entrar los datos, por tanto se propone que se muestre una vista para cada rol que accede al sistema, que contenga solo la información a la que cada uno necesita acceder, evitando así, que se aglomere mucha información en la interfaz con la que va a interactuar cada usuario.

Un papel importante es jugado también por parte de las validaciones de los datos, en cada uno de los módulos, notificando al usuario cada vez que realice una acción indebida, o introduzca datos que no son correctos. También se contempla que cuando los listados estén vacíos se le notifique al usuario mediante un mensaje.

### **1.9. Conclusiones**

En este capítulo se modeló de forma detallada la solución propuesta, a través de los diagramas de clases del análisis y el diseño de cada uno de los casos de uso del sistema. Además se realizaron los modelos lógico y físico de datos de la base de datos propuesta, para responder así a las necesidades de la aplicación, además se detalla cómo se van a tratar los errores en el sistema. Todo esto hace que se concluya la importancia que tiene el análisis y diseño para el desarrollo de cualquier aplicación ya que permite al programador tener una idea clara acerca de lo que se va a hacer para satisfacer las necesidades del cliente.



# CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

## 4.1. Introducción

En este capítulo se desarrollan los flujos de trabajo de implementación y prueba. Se mostrará el modelo de implementación mediante el diagrama de componentes, el cual se realiza según la información que se obtiene del diseño. También se realizará un análisis de los casos de prueba para encontrar los posibles errores que pueda tener el sistema y corregirlos, para así lograr un óptimo funcionamiento del sistema antes de darle uso.

## 4.2. Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos.

Para representar los diagramas del Modelo de Implementación se puede emplear el diagrama de UML de Componentes. (12)

### 4.2.1. Diagrama de componentes

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes de software. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. (12)

A continuación se muestran los diagramas de componentes de una muestra de los casos de uso del sistema que se decidió tomar para plasmar en el documento, el resto se puede encontrar en el Anexo 5.

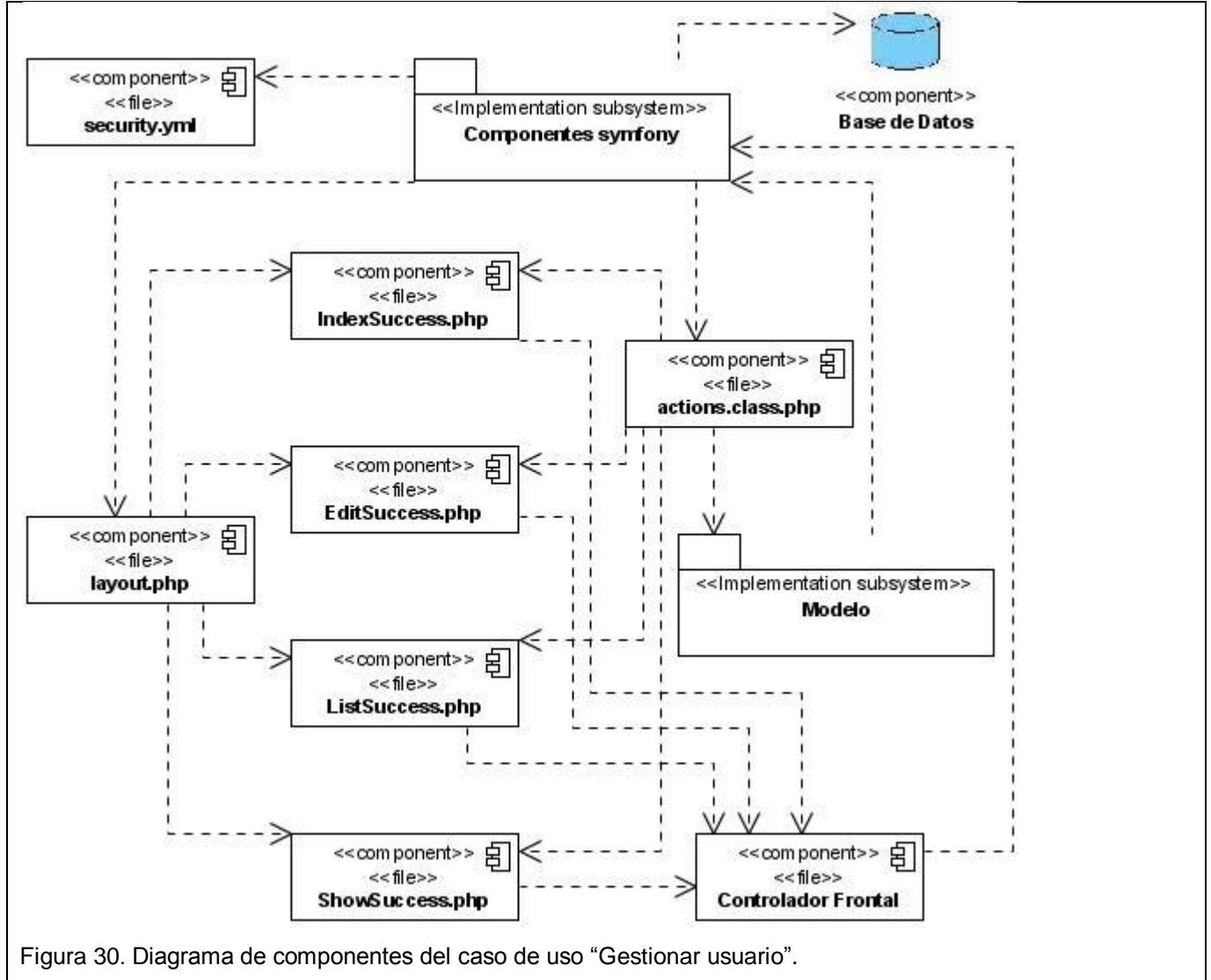


Figura 30. Diagrama de componentes del caso de uso “Gestionar usuario”.

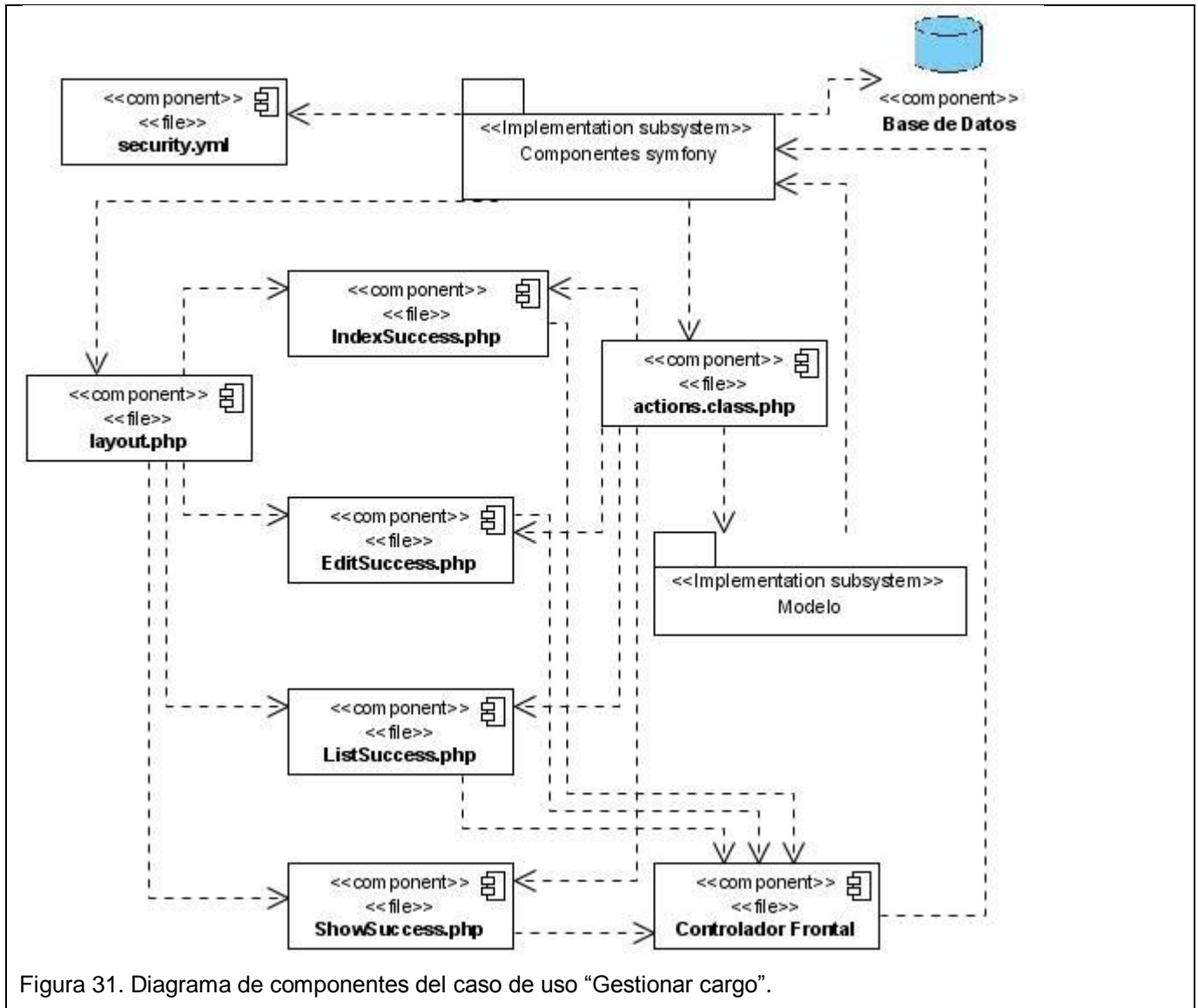


Figura 31. Diagrama de componentes del caso de uso "Gestionar cargo".

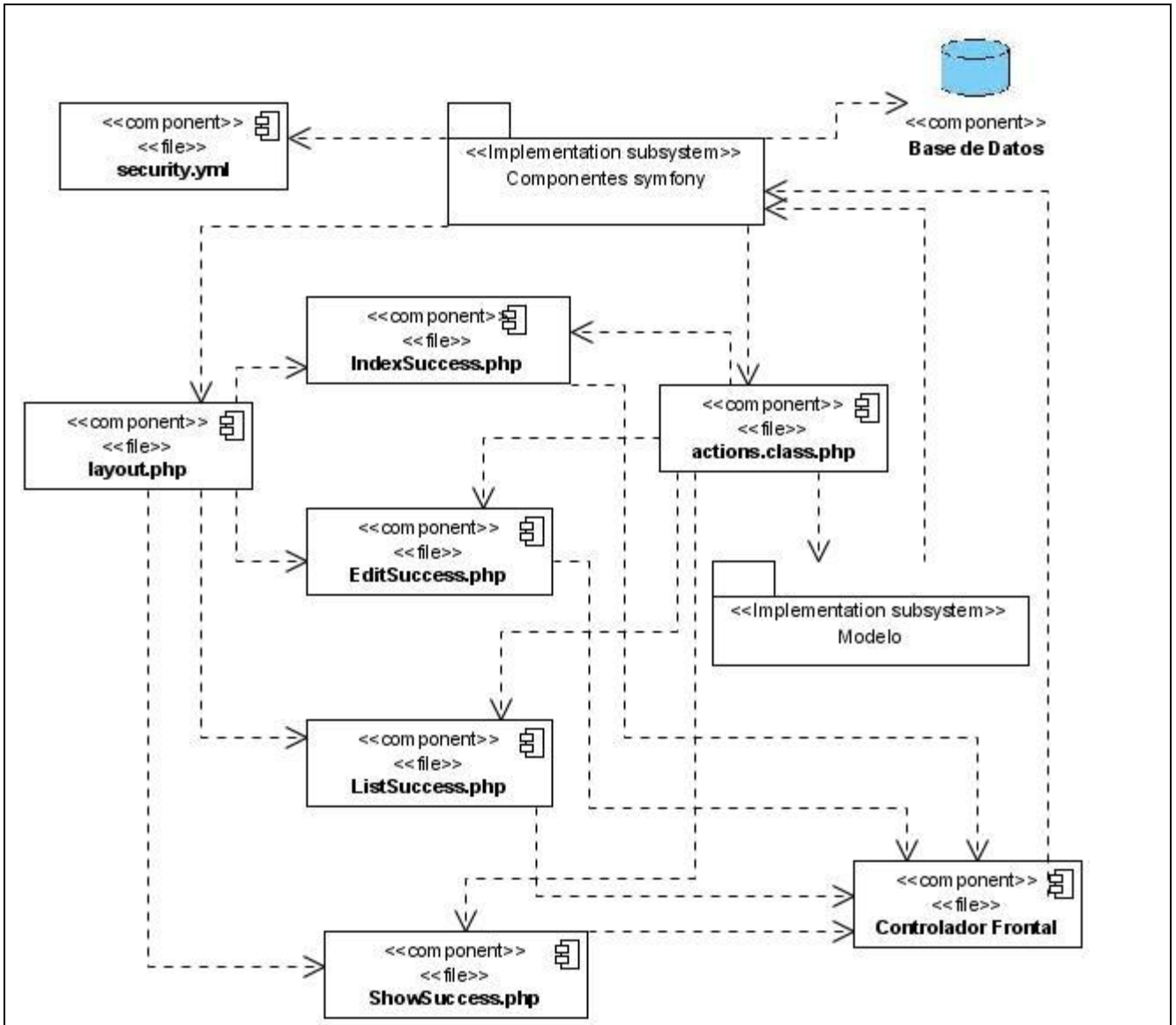


Figura 32. Diagrama de componentes del módulo “Gestionar área”.

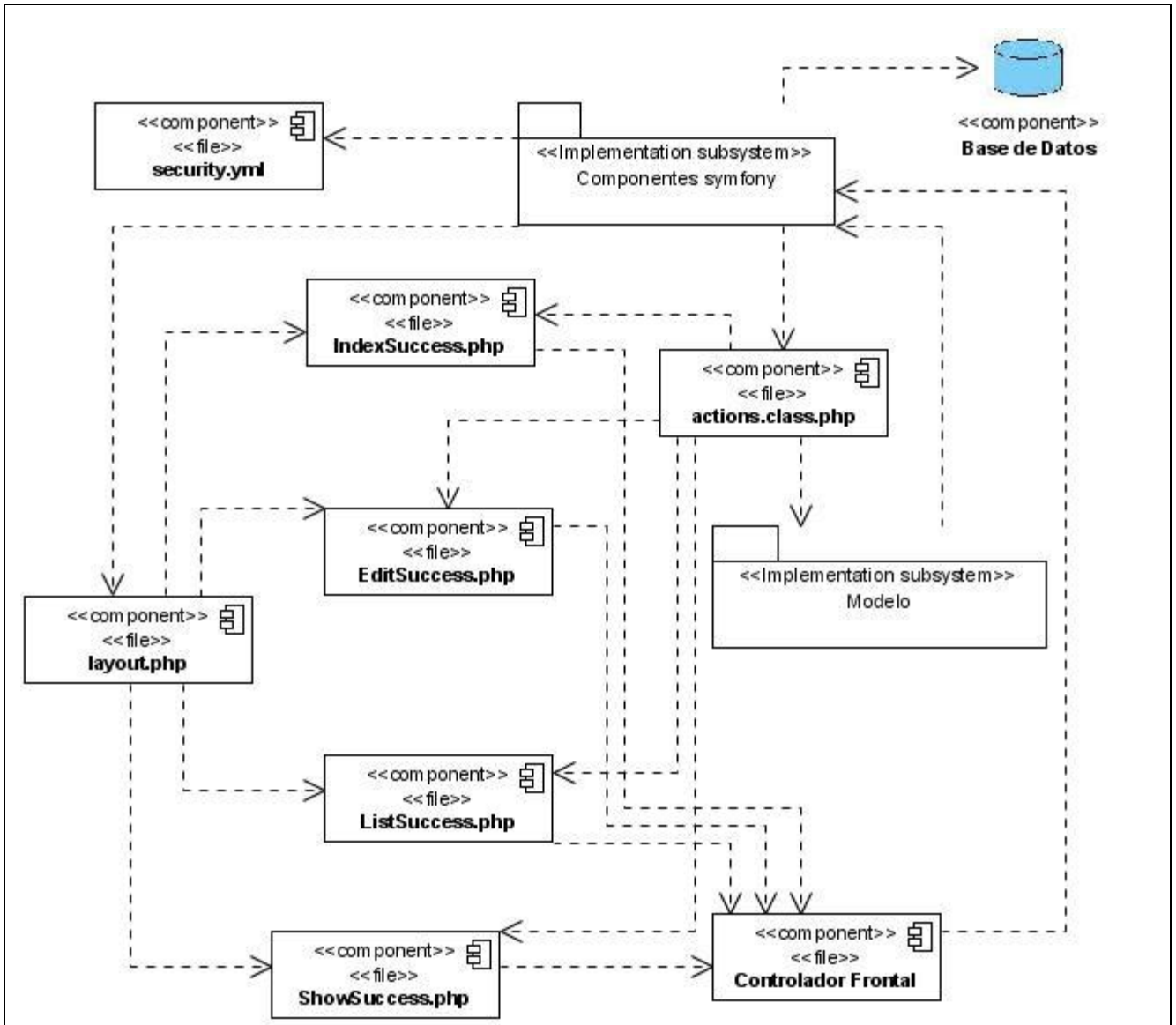


Figura 33. Diagrama de componentes del módulo “Gestionar evaluación”.

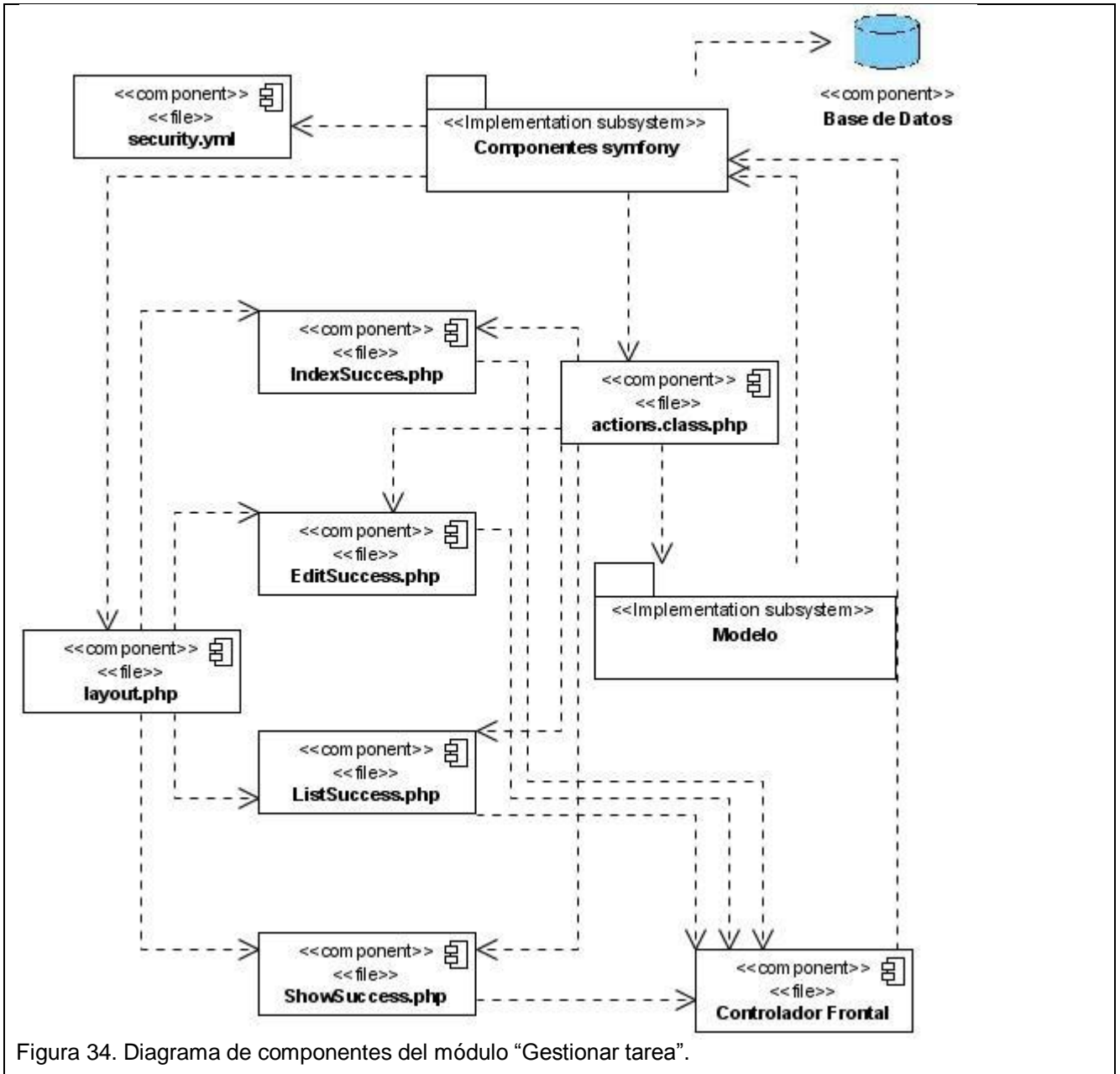


Figura 34. Diagrama de componentes del módulo "Gestionar tarea".

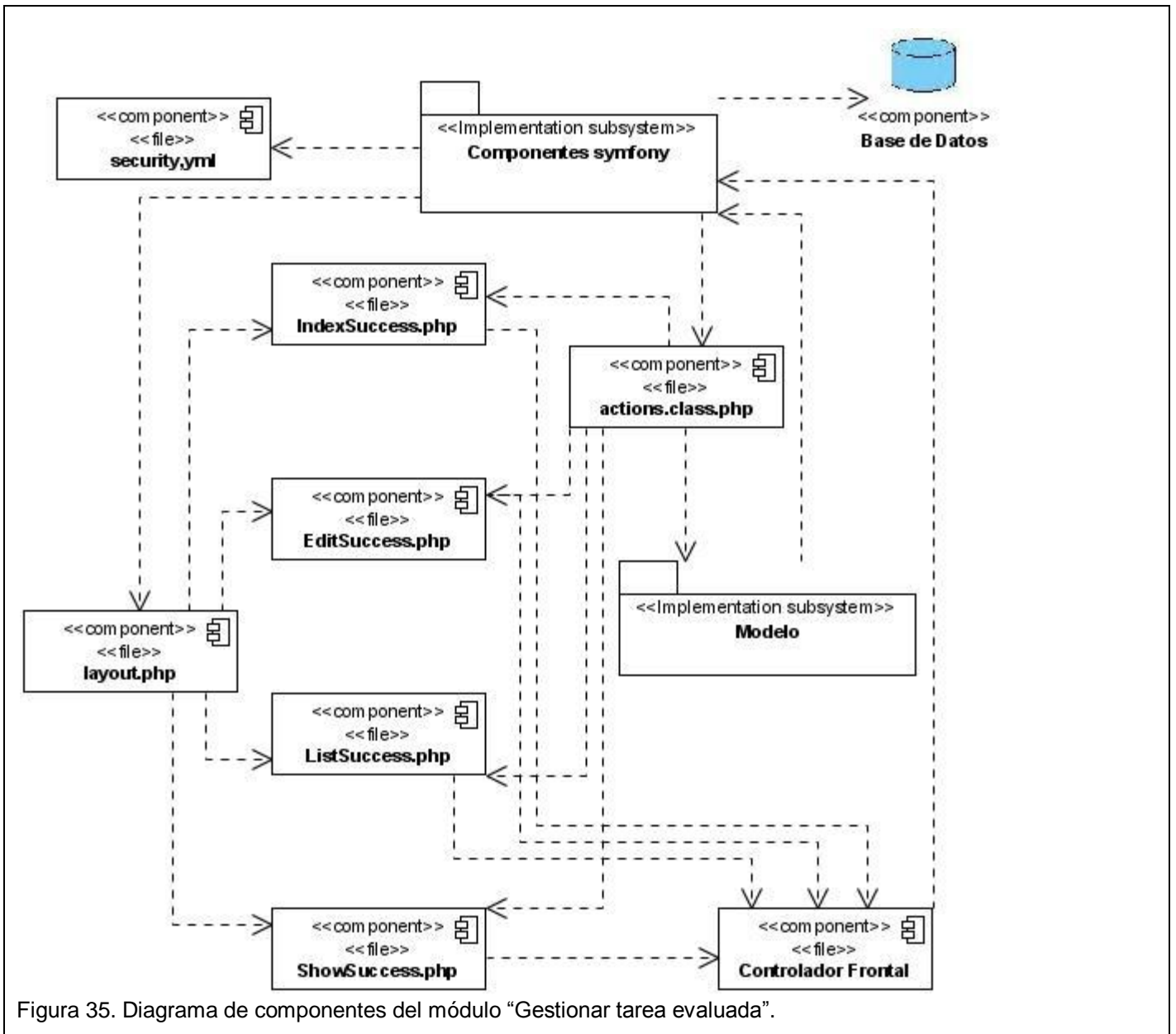


Figura 35. Diagrama de componentes del módulo "Gestionar tarea evaluada".

### 4.3. Marco de trabajo de pruebas

El marco de trabajo Symfony incluye su propio marco de trabajo de pruebas llamado **Lime** que se basa en la librería Test::More de Perl y es compatible con TAP(*Test Anything Protocol*), protocolo creado para facilitar la lectura de los resultados de las pruebas. Está basado en la metodología TDD (desarrollo basado en pruebas) el cual establece que las pruebas se escriben antes que el código de la aplicación. Esto ayuda a pensar y centrarse en el funcionamiento de un método antes de programarlo. Las pruebas unitarias se encargan de un único caso cada vez, lo que significa que un único método puede necesitar varias pruebas unitarias si su funcionamiento varía en función del contexto.

Las pruebas unitarias de Symfony son archivos PHP cuyo nombre termina en Test.php y que se encuentran en el directorio **test/unit/** del proyecto. Su sintaxis es sencilla y fácil de leer.

#### Evaluación de la ejecución del test y de los resultados

Funcionalidad	Método de Prueba	Recibe	Esperado	Regresó
ObtenerUsuariosEvaluadores()	ok		Not ok	Not ok
ObtenerUsuariosEvaluadores()	ok		ok	ok
ObtenerEvaluadosDeUnEvaluador()	ok	2	Not ok	Not ok
ObtenerEvaluadosDeUnEvaluador()	ok	5	ok	ok
LlenarTabla()		is	<u>ok</u>	ok

Tabla 4.1. Pruebas de la clase TbRevaluadorEvaluadoPeer.

Funcionalidad	Método de Prueba	Recibe	Esperado	Regresó
ObtenerUsuariosEvaluados()	ok		Not ok	Not ok
ObtenerUsuariosEvaluados()	ok		ok	ok
ObtenerUsuariosEvaluadoresDadoEvaluado	ok	3	Not ok	Not ok
ObtenerUsuariosEvaluadoresDadoEvaluado	ok	5	ok	ok
LlenarTabla()		is	Ok	ok

Tabla 4.2. Pruebas de la clase TbRtareaEvaluadaPeer.



```

$databaseManager = new sfDatabaseManager();

$databaseManager->initialize();

///*****

$probador= new lime_test(30, new lime_output_color());

$probador->diag('Inicio de prueba para TbRevaluadorEvaluadoPeer////////////////////');

$probador->diag('////////////////////ObtenerUsuariosEvaluadores////////////////////');
$probador->ok( TbRevaluadorEvaluadoPeer::ObtenerUsuariosEvaluadores() == null, 'no existen ');

$probador->diag('////////////////////ObtenerEvaluadosDeUnEvaluador////////////////////');
$evaluador=TbDusuarioPeer::retrieveByPK(2);
$probador->ok(TbRevaluadorEvaluadoPeer::ObtenerEvaluadosDeUnEvaluador($evaluador) != null, 'no existe ');

$probador->diag('//////////Inicio de prueba para TbRtareaEvaluadaPeer//////////');
$probador->diag('//////////Inicio de prueba para TbRtareaEvaluadaPeer//////////');

$probador->diag('//////////ObtenerUsuariosEvaluados//////////');
$probador->ok(bRevaluadorEvaluadoPeer::ObtenerUsuariosEvaluados() == null, 'no existen ');

```

Figura 36. Fragmento de código para las clases TbRevaluadorEvaluadoPeer y TbRtareaEvaluadaPeer.

Funcionalidad	Método de Prueba	Recibe	Esperado	Regresó
LogSoap	ok	50061	ok	ok
LogSoap	ok	dddddd	Not ok	Not ok
Logsoap`	isa_ok	61315	Ok	ok

Tabla 4.3. Pruebas de la clase soap.class.php.

```

72 $probador->diag('Inicio de prueba para soap.class.php////////////////////');
73 $probador->ok(soap::Logsoap('50061')== null, ' existe ');
74 $probador->ok(soap::Logsoap('dddddd')!= null, 'no existe ');
75 $probador->isa_ok('61315', 'string','es un string');
76

```

Figura 37. Fragmento de código para la clase soap.class.php.

Funcionalidad	Método de Prueba	Recibe	Esperado	Regresó
ObtenerEvaluadoresCargo_Area	ok		ok	ok
ObtenerCargoAreaEsEvaluado	ok	1	Not ok	Not ok
ObtenerCargoAreaEsEvaluado	isa_ok	2	Ok	ok

Tabla 4.4. Pruebas de la clase TbRevaluadorEvaluadoCargoAreaPeer.

```

73
74 $probador->diag('Inicio de prueba para TbRevaluadorEvaluadoCargoAreaPeer////////////////////');
75 $probador->ok(TbRevaluadorEvaluadoCargoAreaPeer::ObtenerEvaluadoresCargo_Area== null, 'no existen ');
76 $probador->ok(TbRevaluadorEvaluadoCargoAreaPeer::ObtenerCargoAreaEsEvaluado(1)== null, 'no existe ');
77 $probador->ok(TbRevaluadorEvaluadoCargoAreaPeer::ObtenerCargoAreaEsEvaluado(2)== null, 'no existe ');
78

```

Figura 38. Fragmento de código para la clase TbRevaluadorEvaluadoCargoAreaPeer.

#### 4.4. Descripción de los casos de prueba de integración

A continuación se muestra la descripción de los casos de prueba de algunos de los casos de uso del sistema, el resto se puede encontrar en el anexo 7.

**Nombre del caso de uso:** Gestionar evaluación

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Sus cambios han sido guardados"	El usuario debe estar autenticado en el sistema con el cargo de Webmaster.
El usuario deja el campo vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir evaluación" y da la posibilidad de adicionar la evaluación nuevamente.	

Tabla 4.5. Descripción del caso de prueba Gestionar evaluación. Sección "Adicionar evaluación".

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Sus cambios han sido guardados"	El usuario debe estar autenticado en el sistema con el cargo de Webmaster.
El usuario deja el campo vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir la evaluación" y no se modifican los datos anteriores.	

Tabla 4.6. Descripción del caso de prueba Gestionar evaluación. Sección "Modificar evaluación".

**Nombre del caso de uso:** Gestionar área

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Sus cambios han sido guardados"	El usuario debe estar autenticado en el sistema con el cargo de Webmaster.
El usuario deja el campo vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir el área" y da la posibilidad de adicionarla nuevamente.	

Tabla 4.7. Descripción del caso de prueba Gestionar área. Sección "Adicionar área".

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Sus cambios han sido guardados"	El usuario debe estar autenticado en el sistema con el cargo de Webmaster.
El usuario deja el campo vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir el área" y no se modifican los datos anteriores.	

Tabla 4.8. Descripción del caso de prueba Gestionar área. Sección "Modificar área".

**Nombre del caso de uso:** Gestionar cargo

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Sus cambios han sido guardados"	El usuario debe estar autenticado en el sistema con el cargo de Webmaster.
El usuario deja el campo vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir el cargo" y da la posibilidad de añadirla nuevamente.	Debe haber tipos de evaluación insertados en el sistema.
El usuario no selecciona el tipo de evaluación y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe seleccionar tipo de evaluación" y da la posibilidad de añadirla nuevamente.	

Tabla 4.9. Descripción del caso de prueba Gestionar cargo. Sección "Añadir cargo".

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Sus cambios han sido guardados"	El usuario debe estar autenticado en el sistema con el cargo de Webmaster.
El usuario deja el campo vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir el cargo" y no se modifican los datos anteriores.	Debe haber tipos de evaluación insertados en el sistema.

El usuario no selecciona el tipo de evaluación y presiona el botón <b>Guardar</b> .	Se muestra el mensaje “Debe seleccionar tipo de evaluación” y no se modifican los datos anteriores.	
---	---	--

Tabla 4.10. Descripción del caso de prueba Gestionar cargo. Sección “Modificar cargo”.

**Nombre del caso de uso:** Gestionar usuario

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. “Sus cambios han sido guardados”	El usuario debe estar autenticado en el sistema con el cargo de Webmaster.
El usuario deja el campo vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje “No existe solapín”.	Debe haber cargos y áreas insertados en el sistema.
El usuario no selecciona el área y presiona el botón <b>Guardar</b> .	Se muestra el mensaje “Tiene que seleccionar un área”.	Debe haber cargos asignados a las áreas.

Tabla 4.11. Descripción del caso de prueba Gestionar usuario. Sección “Adicionar usuario”.

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. “Cambios guardados”	El usuario debe estar autenticado en el sistema con el cargo de Webmaster.
El usuario deja el campo vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje “No existe solapín” y no se modifican los datos anteriores.	Debe haber cargos y áreas insertados en el sistema. Debe haber cargos asignados a las áreas.

Tabla 4.12. Descripción del caso de prueba Gestionar usuario. Sección “Modificar usuario”.

**Nombre del caso de uso:** Gestionar tarea evaluada

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado en el sistema con el cargo de Evaluador.
El usuario deja el campo Tarea vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir tarea".	Debe haber usuarios insertados en el sistema. Debe haber tareas insertadas en el sistema.
El usuario no especifica la fecha y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir especificar la fecha de evaluación".	Debe haber evaluaciones insertadas en el sistema. Debe estar definido el evaluador para cada evaluado. Debe haber cargos asignados a las áreas.

Tabla 4.13. Descripción del caso de prueba Gestionar tarea evaluada. Sección "Adicionar tarea evaluada".

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado con el cargo de Evaluador. Debe haber usuarios insertados en el sistema.
El usuario deja el campo Tarea vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir tarea" y se muestra nuevamente la pantalla para crear la tarea.	Debe haber tareas insertadas en el sistema. Debe haber evaluaciones insertadas en el sistema.
El usuario no especifica la fecha y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir especificar la fecha de evaluación" y se muestra nuevamente la pantalla para crear la tarea.	Debe estar definido el evaluador para cada evaluado. Debe haber cargos asignados a las áreas.

Tabla 4.14. Descripción del caso de prueba Gestionar tarea evaluada. Sección "Modificar tarea evaluada".

### 4.5. Conclusiones

En este capítulo se abordó todo lo referente a los flujos de implementación y pruebas. En el caso de la implementación se realizaron los diagramas de componentes, que especifican los componentes usados en cada módulo de la aplicación. En el caso de las pruebas, se realizaron las pruebas unitarias directamente en el código y después de culminar éstas, se realizaron las pruebas de integración, para validar la entrada de datos en el sistema, todo esto permite que la aplicación se encuentre implementada de modo seguro y que los usuarios sean alertados si cometen algún error en la entrada de datos.

## CONCLUSIONES

Con el desarrollo de este trabajo se hace una propuesta de solución al problema de la evaluación por desempeño de los trabajadores en la facultad 1, para lo cual primero se realizó un estudio de los diferentes sistemas que existen a nivel mundial y en nuestro país en lo que a la gestión de la evaluación del desempeño se refiere, llegándose a la conclusión de que ninguno de los sistemas cumple con el reglamento definido por la universidad para la evaluación del desempeño de los trabajadores.

Se realizó el modelado de los procesos del negocio, lo cual permitió un mayor entendimiento acerca del proceso de evaluación, creando así las bases que permitieron obtener un correcto desarrollo del análisis y diseño del sistema. Como principal resultado se obtuvo un sistema informático capaz de llevar a cabo la gestión de toda la información necesaria para la realización de este proceso en la facultad 1, según las normas que establece la UCI, teniendo en cuenta todas las necesidades de los clientes. Esta aplicación puede llegar a ser una importante herramienta para darle soporte a la evaluación del desempeño de los trabajadores en la facultad 1 y realizar este proceso con la eficacia que se necesita en la actualidad.

Para validar la integridad del sistema y su correcto funcionamiento, se realizaron las pruebas unitarias y de integración, obteniendo en todos los casos resultados positivos.

El sistema cumplió con los objetivos propuestos, lográndose así una solución correcta al problema de la gestión de la evaluación laboral. La aplicación resultante cuenta con una interfaz amigable al usuario, la cual posee un diseño uniforme y para la cual se utilizaron tecnología de punta y herramientas modernas de programación orientada a objetos.



## RECOMENDACIONES

Los objetivos generales de este trabajo fueron alcanzados, pero durante su desarrollo, han surgido ideas que sería recomendable tener en cuenta para su futuro perfeccionamiento:

1. Permitir que la aplicación pueda ser utilizada en un futuro no muy lejano, no solo en la facultad 1, sino que se haga extensa a las demás facultades de la universidad.
2. Continuar el desarrollo de este sistema, adicionándole nuevas funcionalidades y servicios que puedan satisfacer necesidades futuras de los clientes, como por ejemplo:
  - Notificación de la evaluación vía correo electrónico.
  - Imprimir certificado de evaluación.
  - Reportes.
3. Realizar una nueva iteración de las funcionalidades y culminar la implementación de la aplicación, además de la confección de un manual de usuario para hacer más fácil su uso.

## BIBLIOGRAFÍA REFERENCIADA

1. Universia. [En línea] [Citado el: 16 de noviembre de 2008.]  
[http://www.universia.cl/portada/actualidad/noticia\\_actualidad.jsp?noticia=139652](http://www.universia.cl/portada/actualidad/noticia_actualidad.jsp?noticia=139652).
2. **Notimex.** Vanguardia. [En línea] [Citado el: 18 de noviembre de 2008.]  
[http://www.vanguardia.com.mx/diario/noticia/estados/nacional/disenan\\_en\\_el\\_ipn\\_sistema\\_informatico\\_para\\_evaluacion\\_laboral/6848](http://www.vanguardia.com.mx/diario/noticia/estados/nacional/disenan_en_el_ipn_sistema_informatico_para_evaluacion_laboral/6848).
3. Wilsoft. [En línea] [Citado el: 18 de noviembre de 2008.] <http://www.wilsoft-la.com/qtraining.htm>.
4. **Dr.Noel Varela Izquierdo, Msc Guillermo A. López, Msc Eduardo Concepción.** Bibliociencias. [En línea] [Citado el: 14 de enero de 2008.]  
<http://www.bibliociencias.cu/gsd/collect/revistas/index/assoc/HASH01f2/d6d2ce0b.dir/doc.pdf>.
5. Sitio de descargas de software. [En línea] [Citado el: 16 de enero de 2009.]  
[http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
6. ZFdes.com. [En línea] [Citado el: 12 de enero de 2009.] <http://manual.zfdes.com/es/introduction.html>.
7. **Valdés, Damián Pérez.** Maestros del Web. [En línea] [Citado el: 14 de diciembre de 2008.]  
<http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>.
8. **Masip, David.** desarrolloweb.com. [En línea] [Citado el: 14 de diciembre de 2009.]  
<http://www.desarrolloweb.com/articulos/840.php>.
9. Entorno Virtual de Aprendizaje. *Conferencia 2. Modelo del negocio.* [En línea] [Citado el: 3 de febrero de 2009.]  
<http://teleformacion.uci.cu/mod/resource/view.php?id=11553>.
10. Entorno Virtual de Aprendizaje. *ISW 1, Conferencia 7, Análisis y Diseño.* [En línea] [Citado el: 18 de febrero de 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=13151>.
11. **Camilo Javier Solís Álvares, Roberth Gustavo Figueroa Díaz.** MYGNET. [En línea] [Citado el: 10 de enero de 2009.] [http://www.mygnet.net/manuales/software/metodologias\\_tradicionales\\_vs\\_dot\\_metodologias\\_agiles.1515](http://www.mygnet.net/manuales/software/metodologias_tradicionales_vs_dot_metodologias_agiles.1515).
12. MeRinde. *Gobierno Bolivariano de Venezuela.* [En línea] [Citado el: 25 de mayo de 2009.]  
[http://merinde.rinde.gob.ve/index.php?option=com\\_content&task=view&id=96&Itemid=297](http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297).

## BIBLIOGRAFÍA CONSULTADA

1. **Dr.Noel Varela Izquierdo, Msc Guillermo A. López, Msc Eduardo Concepción.** Bibliociencias. [En línea] [Citado el: 14 de enero de 2008.]  
<http://www.bibliociencias.cu/gsd/collect/revistas/index/assoc/HASH01f2/d6d2ce0b.dir/doc.pdf>.
2. **Públicas, Ministerio de Administraciones.** Gobierno de España, Ministerio de la Presidencia. [En línea] [Citado el: 10 de enero de 2009.] <http://www.csi.map.es/csi/metrica3/introduccion.pdf>.
3. Entorno Virtual de Aprendizaje. [En línea] [Citado el: 17 de diciembre de 2008.]  
<http://teleformacion.uci.cu/mod/resource/view.php?id=11406>.
4. **Camilo Javier Solís Álvares, Roberth Gustavo Figueroa Díaz.** MYGNET. [En línea] [Citado el: 10 de enero de 2009.]  
[http://www.mygnet.net/manuales/software/metodologias\\_tradicionales\\_vs\\_dot\\_metodologias\\_agiles.1515](http://www.mygnet.net/manuales/software/metodologias_tradicionales_vs_dot_metodologias_agiles.1515).
5. Sitio de descargas de software. [En línea] [Citado el: 16 de enero de 2009.]  
[http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
6. **Valdés, Damián Pérez.** Maestros del Web. [En línea] [Citado el: 19 de enero de 2009.]  
<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
7. ZFdes.com. [En línea] [Citado el: 12 de enero de 2009.] <http://manual.zfdes.com/es/introduction.html>.
8. **Valdés, Damián Pérez.** Maestros del Web. [En línea] [Citado el: 14 de diciembre de 2008.]  
<http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>.
9. MySQL. [En línea] [Citado el: 14 de diciembre de 2008.]  
<http://dev.mysql.com/doc/refman/5.0/es/introduction.html>.
10. **Masip, David.** desarrolloweb.com. [En línea] [Citado el: 14 de diciembre de 2009.]  
<http://www.desarrolloweb.com/articulos/840.php>.
11. **Martínez, Rafael.** PostgreSQL-es-org. [En línea] [Citado el: 15 de enero de 2009.]  
<http://www.desarrolloweb.com/articulos/840.php>.
12. Entorno Virtual de Aprendizaje. *Conferencia 2. Modelo del negocio.* [En línea] [Citado el: 3 de febrero de 2009.]  
<http://teleformacion.uci.cu/mod/resource/view.php?id=11553>.

13. **Centro Nacional de Tecnologías de Información.** Gobierno Bolivariano de Venezuela. [En línea] [Citado el: 12 de febrero de 2009.] [http://merinde.rinde.gob.ve/index.php?option=com\\_content&task=view&id=128&Itemid=294](http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=128&Itemid=294).
14. Entorno Virtual de Aprendizaje. *ISW 1, Conferencia 7, Análisis y Diseño.* [En línea] [Citado el: 18 de febrero de 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=13151>.
15. MeRinde. *Gobierno Bolivariano de Venezuela.* [En línea] [Citado el: 25 de mayo de 2009.] [http://merinde.rinde.gob.ve/index.php?option=com\\_content&task=view&id=96&Itemid=297](http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297).
16. Entorno Virtual de Aprendizaje. *Ingeniería de Software 2/ Conferencia Implementacion.* [En línea] [Citado el: 24 de mayo de 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=14094>.
17. **Fabien Potencier, François Zaninotto.** *Symfony, la guía definitiva.*
18. **Mara Julia Leandro Baños, Leyanis Santiesteban Quintana, Surayne Torres López.** *Sistema de Gestión de la Ubicación Laboral de los Egresados de la UCI.* Ciudad de la Habana : s.n., 2008.
19. **Marta Alvarado Garrigó, Nelvis Borges Fernández.** *Sistema Informático para la Red Nacional de Genética Médica: módulo Teleconsulta versión 2.0.* Ciudad de la Habana : s.n., 2008.
20. **Damián Cervantes Rodón, Rolando Ávila Figueredo.** *Análisis y diseño del sistema de venta y admisión de la plataforma de servicios postales.* Ciudad de la Habana : s.n., 2007.
21. **Velázquez, Catherine Muñoz.** *Análisis y Diseño del Subsistema Matrícula. Sistema Automatizado para la Gestión Académica – “Akademos 2.0”.* Ciudad de la Habana : s.n., 2008.
22. **S.Pressman, Roger.** *Ingeniería del software. Un enfoque práctico.* 2001.
23. **Meléndrez, Edelsys Hernández.** *Metodología de la investigación. Como escribir una tesis.* 2006.
24. Mis algoritmos. [En línea] [Citado el: 28 de abril de 2009.] <http://mis-algoritmos.com/seguridad-en-aplicaciones-web>.
25. **Equipo de desarrollo de PostgreSQL.** ABCdatos. [En línea] [Citado el: 14 de enero de 2009.] <http://descargas.abcdatos.com/tutorial/descargarL6789.html>.
26. Universia. [En línea] [Citado el: 16 de noviembre de 2008.] [http://www.universia.cl/portada/actualidad/noticia\\_actualidad.jsp?noticia=139652](http://www.universia.cl/portada/actualidad/noticia_actualidad.jsp?noticia=139652).
27. **Notimex.** Vanguardia. [En línea] [Citado el: 18 de noviembre de 2008.] [http://www.vanguardia.com.mx/diario/noticia/estados/nacional/disenan\\_en\\_el\\_ipn\\_sistema\\_informatico\\_para\\_evaluacion\\_laboral/6848](http://www.vanguardia.com.mx/diario/noticia/estados/nacional/disenan_en_el_ipn_sistema_informatico_para_evaluacion_laboral/6848).

28. Wilsoft. [En línea] [Citado el: 18 de noviembre de 2008.] <http://www.wilsoft-la.com/qtraining.htm>.

**Valdés, Damián Pérez.** Maestros del Web. [En línea] [Citado el: 19 de enero de 2009.]  
<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.

**Velázquez, Catherine Muñoz.** *Análisis y Diseño del Subsistema Matrícula. Sistema Automatizado para la Gestión Académica – “Akademos 2.0”*. Ciudad de la Habana : s.n., 2008.

---

## GLOSARIO DE TÉRMINOS

1. **DOM: (*Document Object Model*)**. Conjunto de utilidades específicamente diseñadas para manipular documentos XML. Por extensión, DOM también se puede utilizar para manipular documentos XHTML y HTML.
2. **HTTP: (*Hypertext Transfer Protocol*)** Protocolo de transmisión de hipertexto.
3. **PHP: (*Hypertext Pre-processor*)** Es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es un acrónimo recurrente que significa Preprocesador de Hipertexto PHP.
4. **SQL: (*Structured Query Language*)** Lenguaje de Consulta Estructurado, es un lenguaje declarativo de acceso a bases de datos relacionales
5. **Servicio Web: (*Web Service*)** Es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
6. **HTML: (*HyperText Markup Language*)**. Lenguaje de Marcas de Hipertexto, es el lenguaje de marcado predominante para la construcción de páginas web.
7. **XML: (*Extensible Markup Language*)** un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium.
8. **WYSIWYG (*What You See Is What You Get*)**. Lo que ves es lo que obtienes, se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML)
9. **Herramienta CASE (*Computer Aided Software Engineering*)**. Ingeniería de Software Asistida por Ordenador son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software.
10. **XSS (*Cross-Site Scripting*)**. Es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado.
11. **CGI (*Common Gateway Interface*)**. Interfaz de entrada común es una tecnología que se usa en los servidores web.

- 12. ABC** es un lenguaje de programación de alto nivel. Inicialmente fue pensado como sustituto del lenguaje BASIC.
- 13. Servicio Web: (*Web Service*)** Es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
- 14. YAML: (*Ain't Markup Language*)** Lenguaje de serialización diseñado a partir de la estructura de datos más comunes de los lenguajes de programación ágiles. Es muy útil para las necesidades de configuración.

## ANEXOS

### Anexo 1: Diagramas de actividades del negocio.

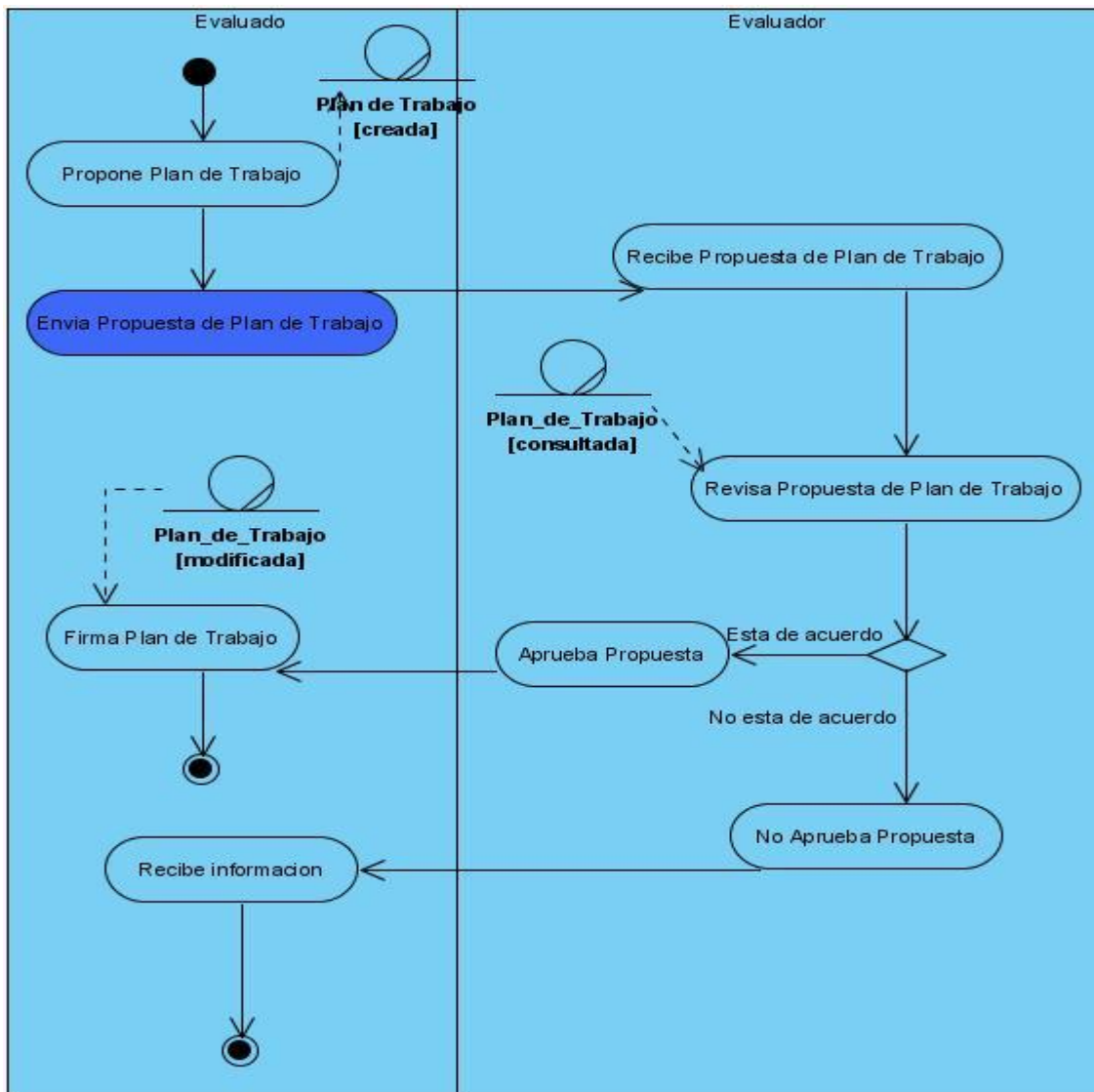


Diagrama de actividades del caso de uso "Agregar plan de trabajo"



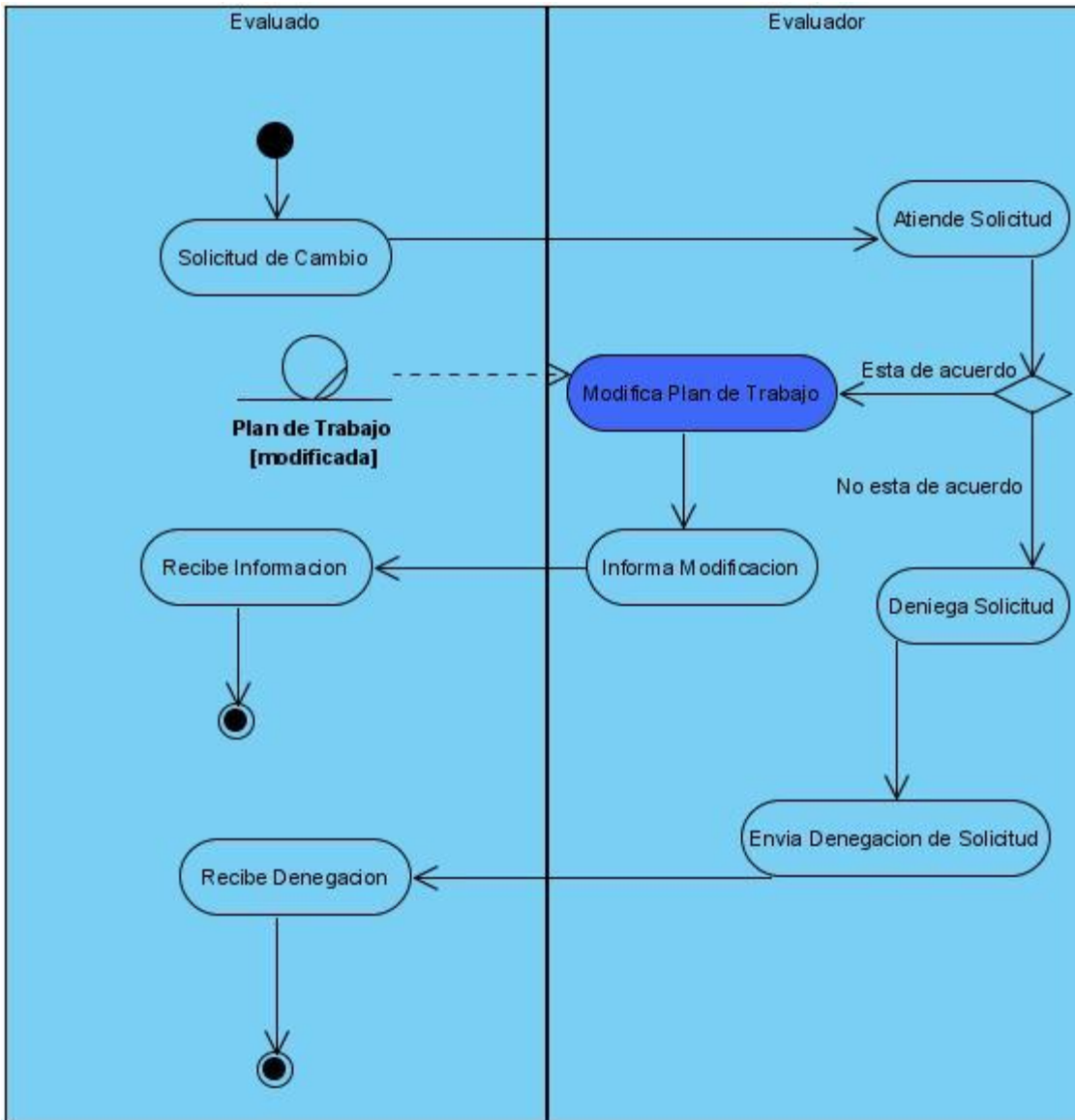


Diagrama de actividades del caso de uso “Modificar plan de trabajo”

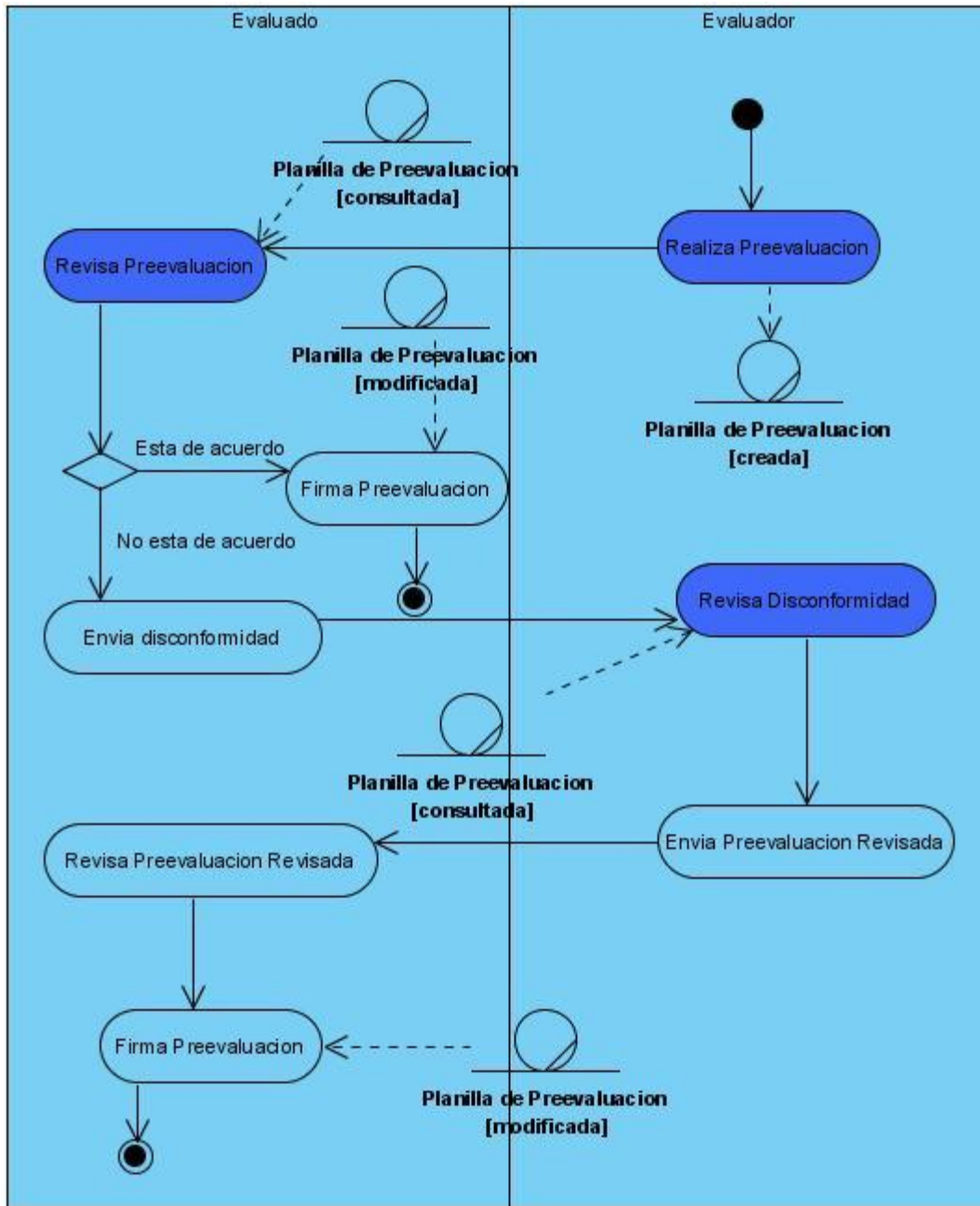
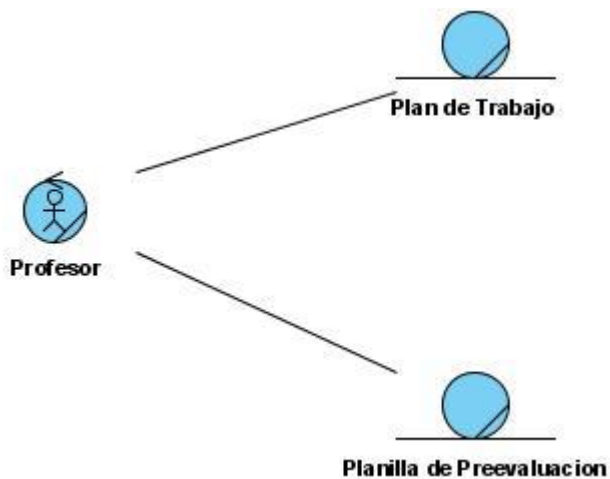



Diagrama de actividades del caso de uso “Realizar evaluación”




**Anexo 2: Diagrama de objetos del negocio.**




**Anexo 3: Descripción extendida de los casos de uso del sistema.**

CU_1	Gestionar Evaluación
Actor	Administrador
Propósito	Permitir al administrador gestionar las evaluaciones de las tareas que realizan los usuarios del sistema.
Descripción	El caso de uso se inicia cuando el administrador solicita insertar una evaluación, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.
Prioridad	Crítica
Escenario 1: Adicionar Evaluación	
Curso normal de los eventos	
Actor	Respuesta del sistema



<p>1. El administrador selecciona la opción adicionar.</p>	<p>2. Se muestra un formulario a llenar y se le da la posibilidad al administrador de crear una evaluación.          3. Se validan los datos entrados.          4. Se almacena la evaluación adicionada          5. Se notifica que se creó la evaluación y se finaliza en caso de uso.</p>
<p>Flujo alternativo 2</p>	
	<p>2.1. Si la evaluación ya existe el sistema lo notifica y da la posibilidad de crear una nueva.</p>
<p>Prototipo de Interfaz</p>	
	
<p>Escenario 2: Modificar Evaluación</p>	
<p>Curso normal de los eventos</p>	
<p>Actor</p>	<p>Respuesta del sistema</p>
<p>1. El administrador selecciona la opción modificar.</p>	<p>2. Se muestra un formulario con todas las evaluaciones creadas.          3. Se muestra un formulario con todos los datos modificables.          4. Se almacenan los datos y finaliza el caso de uso.</p>
<p>Flujo alternativo 2</p>	
	<p>2.1. Si no existen evaluaciones en el sistema no se permiten cambios.</p>
<p>Flujo alternativo 3</p>	
	<p>3.1. Si se modifican los datos el sistema lo notifica.</p>
<p>Prototipo de Interfaz</p>	


	
Escenario 3: Eliminar Evaluación	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El actor selecciona la opción eliminar	2. El sistema permite eliminar la evaluación seleccionada. 3. Se notifica que la evaluación se eliminó satisfactoriamente.
Flujo alternativo 2	
	2.1. Se muestra un mensaje de confirmación antes de eliminar definitivamente.
Prototipo de Interfaz	
<h3>Tipos de desempeños laborales</h3> <div style="display: flex; justify-content: space-around; align-items: center;"> <div data-bbox="462 1024 760 1365">  </div> <div data-bbox="836 1045 1209 1297">  </div> </div>	
Escenario 4: Mostrar Evaluación	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El usuario solicita gestionar evaluación.	2. El sistema debe mostrar un formulario con todas las evaluaciones que se hayan creado.
Flujo alternativo 2	

	3.1. Si no existen evaluaciones, el sistema debe mostrar un mensaje notificando que el listado está vacío.
Prototipo de Interfaz	
	

Descripción extendida del caso de uso “Gestionar evaluación”

CU_2	Gestionar Área
Actor	Administrador
Propósito	Permitir al administrador gestionar las áreas a las que van a estar asignados los usuarios del sistema.
Descripción	El caso de uso se inicia cuando el administrador solicita insertar un área, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.
Prioridad	Crítica
Escenario 1: Adicionar Área	
Curso normal de los eventos	
Actor	Respuesta del sistema

<p>1. El administrador selecciona la opción adicionar.</p>	<p>2. Se muestra un formulario a llenar y se le da la posibilidad al administrador de crear un área.          3. Se validan los datos entrados.          4. Se almacena la evaluación adicionada          5. Se notifica que se creó el área y se finaliza en caso de uso.</p>
<p>Flujo alterno 2</p>	
	<p>2.1. Si el área ya existe el sistema lo notifica y da la posibilidad de crear una nueva.</p>
<p>Prototipo de Interfaz</p>	
	
<p>Escenario 2: Modificar Área</p>	
<p>Curso normal de los eventos</p>	
<p>Actor</p>	<p>Actor</p>
<p>1. El administrador selecciona la opción modificar.</p>	<p>2. Se muestra un formulario con todas las áreas creadas.          3. Se muestra un formulario con todos los datos modificables.          4. Se almacenan los datos y finaliza el caso de uso.</p>
<p>Flujo alterno 3</p>	
	<p>3.1. Si se modifican los datos el sistema lo notifica.</p>
<p>Prototipo de Interfaz</p>	
	
<p>Escenario 3: Eliminar Área</p>	
<p>Curso normal de los eventos</p>	



Actor		Actor	
1.	El actor selecciona la opción eliminar.	2.	El sistema permite eliminar el área seleccionada.
		3.	Se notifica que el área se eliminó satisfactoriamente.
Flujo alternativo 2			
		2.1.	Se muestra un mensaje de confirmación antes de eliminar definitivamente.
Prototipo de Interfaz			
 <p>The screenshot shows a web interface titled "Listado de Áreas". It contains a table with four rows: "Administracion del sitio", "DECANATO", "DEPARTAMENTO DE SISTEMAS DIGITALES", and "VICEDECANATO DE PRODUCCION". Each row has three icons: a document with a checkmark, a document with a magnifying glass, and a document with a person icon. Below the table is a blue plus icon. To the right of the table is a "Windows Internet Explorer" dialog box titled "Desea Eliminar" with "Aceptar" and "Cancelar" buttons.</p>			
Escenario 4: Mostrar Área			
Curso normal de los eventos			
Actor		Actor	
1.	El usuario solicita gestionar áreas.	2.	El sistema debe mostrar un formulario con todas las áreas que se hayan creado.
Flujo alternativo 2			
		1.1.	Si no existen áreas, el sistema debe mostrar un mensaje notificando que el listado está vacío.





Descripción extendida del caso de uso “Gestionar área”


CU_3	Gestionar Cargo	
Actor	Administrador	
Propósito	Permitir al administrador gestionar los cargos que tendrán los usuarios del sistema.	
Descripción	El caso de uso se inicia cuando el administrador solicita insertar un cargo, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.	
Prioridad	Crítica	
Escenario 1: Adicionar Cargo		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción adicionar.	2. Se muestra un formulario a llenar y se le da la posibilidad al administrador de crear un cargo. 3. Se validan los datos entrados. 4. Se almacena el cargo adicionado. 5. Se notifica que se creó el cargo y se finaliza en caso de uso.	

Flujo alternativo 2	
	2.1. Si el cargo ya existe el sistema lo notifica y da la posibilidad de crear uno nuevo.
Prototipo de Interfaz	
	
Escenario 2: Modificar Cargo	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El administrador selecciona la opción modificar.	2. Se muestra un formulario con todos los cargos creados. 3. Se muestra un formulario con todos los datos modificables. 4. Se almacenan los datos y finaliza el caso de uso.
Flujo alternativo 3	
	3.1. Si se modifican los datos el sistema lo notifica.
Prototipo de Interfaz	
	
Escenario 3: Eliminar Cargo	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El actor selecciona la opción eliminar	2. El sistema permite eliminar el cargo seleccionado. 3. Se notifica que el cargo se eliminó satisfactoriamente.
Flujo alternativo 2	
	2.1. Se muestra un mensaje de confirmación antes de eliminar definitivamente.

Prototipo de Interfaz	
	
Escenario 3: Mostrar Cargo	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El usuario solicita gestionar áreas.	2. El sistema debe mostrar un formulario con todos los cargos que se hayan creado.
Flujo alternativo 2	
	1.1. Si no existen áreas, el sistema debe mostrar un mensaje notificando que el listado está vacío.
Prototipo de Interfaz	
	

Descripción extendida del caso de uso "Gestionar cargo"

CU_4	Gestionar Usuario	
Actor	Administrador	
Propósito	Permitir al administrador gestionar los usuarios del sistema.	
Descripción	El caso de uso se inicia cuando el administrador solicita insertar un usuario, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.	
Prioridad	Crítica	
Escenario 1: Adicionar Usuario		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción adicionar.	2. Se muestra un formulario a llenar y se le da la posibilidad al administrador de crear un usuario. 3. Se validan los datos entrados. 4. Se almacena el cargo adicionado. 5. Se notifica que se creó el usuario y se finaliza en caso de uso.	
Flujo alterno 2		
	2.1. Si el usuario ya existe el sistema lo notifica y da la posibilidad de crear uno nuevo.	

Prototipo de Interfaz																			
<p><b>Nuevo usuario</b></p> <table border="1"> <thead> <tr> <th>Cargo Real:</th> <th>Solapin:</th> <th>Es webmaster:</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/> Webmaster</td> <td>Ver áreas</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/> Decano</td> <td>Ver áreas</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/> Vicedecano</td> <td>Ver áreas</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/> Jefe de departamento</td> <td>Ver áreas</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/> Profesor</td> <td>Ver áreas</td> </tr> </tbody> </table> <p><u>Cuenta Local</u></p> 		Cargo Real:	Solapin:	Es webmaster:	<input type="radio"/>	<input type="checkbox"/> Webmaster	Ver áreas	<input type="radio"/>	<input type="checkbox"/> Decano	Ver áreas	<input type="radio"/>	<input type="checkbox"/> Vicedecano	Ver áreas	<input type="radio"/>	<input type="checkbox"/> Jefe de departamento	Ver áreas	<input type="radio"/>	<input type="checkbox"/> Profesor	Ver áreas
Cargo Real:	Solapin:	Es webmaster:																	
<input type="radio"/>	<input type="checkbox"/> Webmaster	Ver áreas																	
<input type="radio"/>	<input type="checkbox"/> Decano	Ver áreas																	
<input type="radio"/>	<input type="checkbox"/> Vicedecano	Ver áreas																	
<input type="radio"/>	<input type="checkbox"/> Jefe de departamento	Ver áreas																	
<input type="radio"/>	<input type="checkbox"/> Profesor	Ver áreas																	
Escenario 2: Modificar Usuario																			
Curso normal de los eventos																			
Actor	Respuesta del sistema																		
1. El administrador selecciona la opción modificar.	2. Se muestra un formulario con todos los usuarios creados. 3. Se muestra un formulario con todos los datos modificables. 4. Se almacenan los datos y finaliza el caso de uso.																		
Flujo alterno 3																			
	1.1. Si se modifican los datos el sistema lo notifica.																		

Prototipo de Interfaz										
										
<h3>Reynaldo Lázaro Batista Vázquez</h3>										
<p>Es administrador: <input type="checkbox"/></p>										
<p>Cargo Real: Solapin: <input type="text" value="50061"/></p>										
<table border="1"> <tr> <td><input checked="" type="radio"/></td> <td><input checked="" type="checkbox"/> Webmaster</td> <td>Ver areas</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/> JEFE DE DEPARTEMENTO</td> <td>Ver areas</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/> PROFESOR</td> <td>Ver areas</td> </tr> </table>		<input checked="" type="radio"/>	<input checked="" type="checkbox"/> Webmaster	Ver areas	<input type="radio"/>	<input type="checkbox"/> JEFE DE DEPARTEMENTO	Ver areas	<input type="radio"/>	<input type="checkbox"/> PROFESOR	Ver areas
<input checked="" type="radio"/>	<input checked="" type="checkbox"/> Webmaster	Ver areas								
<input type="radio"/>	<input type="checkbox"/> JEFE DE DEPARTEMENTO	Ver areas								
<input type="radio"/>	<input type="checkbox"/> PROFESOR	Ver areas								
<p><u>Cuenta Local</u></p>										
										
Escenario 3: Eliminar Usuario										
Curso normal de los eventos										
Actor	Actor									
1. El actor selecciona la opción eliminar.	2. El sistema permite eliminar el usuario seleccionado. 3. Se notifica que el usuario se eliminó satisfactoriamente.									
Flujo alternativo 2										
	2.1. Se muestra un mensaje de confirmación antes de eliminar definitivamente.									

Prototipo de Interfaz				
<b>Listado de usuarios</b>				
Foto	Nombre	Área	Webmaster	Acciones
	Reynaldo Lázaro Batista Vázquez	Webmaster del Sistema		 
	FRANCISCO JAVIER HERNANDEZ CAO	D		 
	MARIANNY HERNANDEZ BATISTA	V p	e	 
	Lilianny Marrero Travieso	Webmaster del Sistema		 
				
<b>Escenario 4: Mostrar Usuario</b>				
Curso normal de los eventos				
Actor		Actor		
1. El usuario solicita gestionar usuario.		2. El sistema muestra el listado de usuarios existentes.		
Flujo alternativo 2				
		2.1. Si no existen usuarios, el sistema debe mostrar un mensaje notificando que el listado está vacío.		


Prototipo de Interfaz				
Listado de usuarios				
Foto	Nombre	Área	Webmaster	Acciones
	Reynaldo Lázaro Batista Vázquez	Webmaster del Sistema		 
	FRANCISCO JAVIER HERNANDEZ CAO	Decano del Decanato		 
	MARIANNY HERNANDEZ BATISTA	Vicedecano del Vicedecanato de producción		 
	Lilianny Marrero Travieso	Webmaster del Sistema		 
				



Descripción extendida del caso de uso "Gestionar usuario"




CU_5	Crear cuenta local
Actor	Administrador
Propósito	Permitir al administrador gestionar los usuarios locales del sistema.
Descripción	El caso de uso se inicia cuando el administrador solicita realizar una de las funcionalidades de gestión a los usuarios del sistema, el sistema le da la posibilidad de crear una cuenta local para cada usuario.
Prioridad	Crítica
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El administrador selecciona la opción adicionar.	2. El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al usuario de crear una cuenta local. 3. Se validan los datos. 4. Se almacenan los datos.
Flujo alternativo 2	
	2.1. Si el usuario ya existe se notifica y se da la posibilidad de crear el usuario nuevamente.
Prototipo de Interfaz	

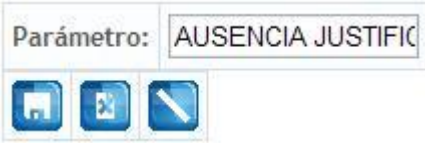
Descripción extendida del caso de uso “Crear cuenta local”

CU_6	Gestionar Tipo de Tarea	
Actor	Administrador	
Propósito	Permitir al administrador gestionar los tipos de tareas que van a existir en el sistema.	
Descripción	El caso de uso se inicia cuando el administrador solicita insertar un tipo de tarea, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.	
Prioridad	Crítica	
Escenario 1: Adicionar Tipo de Tarea		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción adicionar.	2. Se muestra un formulario a llenar y se le da la posibilidad al administrador de crear un tipo de tarea. 3. Se validan los datos entrados. 4. Se almacena el tipo de tarea adicionado. 5. Se notifica que se creó el tipo de tarea y se finaliza en caso de uso.	
Flujo alterno 2		
	2.1. Si el tipo de tarea ya existe el sistema lo notifica y da la posibilidad de crear uno nuevo.	
Prototipo de Interfaz		
		
Escenario 2: Modificar Tipo de Tarea		
Curso normal de los eventos		
Actor	Respuesta del sistema	

<p>1. El administrador selecciona la opción modificar.</p>	<p>2. Se muestra un formulario con todos los tipos de tarea creados.          3. Se muestra un formulario con todos los datos modificables.          4. Se almacenan los datos y finaliza el caso de uso.</p>
<p>Flujo alternativo 3</p>	
	<p>3.1. Si se modifican los datos el sistema lo notifica.</p>
<p>Prototipo de Interfaz</p>	
	
<p>Escenario 3: Eliminar Tipo de Tarea</p>	
<p>Curso normal de los eventos</p>	
<p>Actor</p>	<p>Respuesta del sistema</p>
<p>1. El actor selecciona la opción eliminar</p>	<p>2. El sistema permite eliminar el tipo de tarea seleccionado.          3. Se notifica que el tipo de tarea se eliminó satisfactoriamente.</p>
<p>Flujo alternativo 2</p>	
	<p>2.1. Se muestra un mensaje de confirmación antes de eliminar definitivamente.</p>
<p>Prototipo de Interfaz</p>	
	

Descripción extendida del caso de uso “Gestionar tipo tarea”

CU_7	Gestionar Parámetro	
Actor	Administrador	
Propósito	Permitir al administrador gestionar los parámetros que se vayan a medir en tareas que sean especiales. Por ejemplo: <ul style="list-style-type: none"> <li>• Asistencia</li> <li>• Puntualidad</li> </ul>	
Descripción	El caso de uso se inicia cuando el administrador solicita insertar un usuario, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.	
Prioridad	Crítica	
Escenario 1: Adicionar Parámetro		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción adicionar.	2. Se muestra un formulario a llenar y se le da la posibilidad al administrador de crear un parámetro. 3. Se validan los datos entrados. 4. Se almacena el parámetro adicionado. 5. Se notifica que se creó el parámetro y se finaliza en caso de uso.	
Flujo alterno 2		
	Si el parámetro ya existe el sistema lo notifica y da la posibilidad de crear uno nuevo.	
Prototipo de Interfaz		
		

Escenario 2: Modificar Parámetro	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El administrador selecciona la opción modificar.	2. Se muestra un formulario con todos los parámetros creados. 3. Se muestra un formulario con todos los datos modificables. 4. Se almacenan los datos y finaliza el caso de uso.
Flujo alternativo 3	
	3.1. Si se modifican los datos el sistema lo notifica.
Prototipo de Interfaz	
	
Escenario 3: Eliminar Parámetro	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El actor selecciona la opción eliminar	2. El sistema permite eliminar el parámetro seleccionado. 3. Se notifica que el parámetro se eliminó satisfactoriamente.
Flujo alternativo 2	
	2.1. Se muestra un mensaje de confirmación antes de eliminar definitivamente.



Descripción extendida del caso de uso "Gestionar parámetro"

CU_8	Gestionar Tarea Evaluada	
Actor	Evaluador	
Propósito	Permitir al administrador gestionar las tareas que se les asignarán a los usuarios del sistema.	
Descripción	El caso de uso se inicia cuando el administrador solicita insertar una tarea evaluada, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.	
Prioridad	Crítica	
Escenario 1: Adicionar Tarea Evaluada		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción adicionar.	2. Se muestra un listado con las tareas asignadas al cargo y al área, el administrador puede crear una tarea evaluada. 3. Se validan los datos entrados. 4. Se almacena la tarea evaluada adicionada 5. Se notifica que se creó la tarea evaluada y se finaliza en caso de uso.	
Flujo alterno 2		
	2.1. Si la tarea evaluada ya existe el sistema lo notifica y da la posibilidad de crear una nueva.	
Prototipo de Interfaz		



**EDISTIO YOEL VERDECIA MARTINEZ**

*JEFE DE DEPARTAMENTO del DEPARTAMENTO DE PROGRAMACION*

Nueva Tarea:	<input type="text"/>
Tipo de tarea:	DOCENTE ▾
Evaluacion:	SUPERIOR ▾
Fecha de evaluacion:	<input type="text"/> ...
¿ Es tarea extra?	<input type="checkbox"/>
Observaciones	
<input type="text"/>	
Path:	



**Escenario 2: Modificar Tarea Evaluada**




**Curso normal de los eventos**

Actor	Respuesta del sistema
1. El administrador selecciona la opción modificar.	2. Se muestra un formulario con todas las tareas evaluadas creadas. 3. Se muestra un formulario con todos los datos modificables. 4. Se almacenan los datos y finaliza el caso de uso.





Actor	Respuesta del sistema														
1. El actor selecciona la opción eliminar	2. El sistema permite eliminar la tarea seleccionada. 3. Se notifica que la tarea se eliminó satisfactoriamente.														
Flujo alternativo 2															
	2.1. Se muestra un mensaje de confirmación antes de eliminar definitivamente.														
Prototipo de Interfaz															
<div style="text-align: center;">  <p><b>EDISTIO YOEL VERDECIA MARTINEZ</b> JEFE DE DEPARTAMENTO del DEPARTAMENTO DE PROGRAMACION</p> </div> <h3>Listado de tareas a evaluar</h3> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div data-bbox="289 1016 992 1339" style="width: 60%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="background-color: #e1eef6;">Tarea: Dar Clases</th> </tr> </thead> <tbody> <tr> <td style="background-color: #e1eef6;">Evaluación:</td> <td>ADECUADO</td> </tr> <tr> <td style="background-color: #e1eef6;">Fecha:</td> <td>2009-05-30 00:00:00</td> </tr> <tr> <th colspan="2" style="background-color: #e1eef6;">Observaciones de la tarea</th> </tr> <tr> <td colspan="2" style="text-align: center;">No tiene observaciones</td> </tr> <tr> <td colspan="2">Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO</td> </tr> <tr> <th colspan="2" style="background-color: #e1eef6;">Tarea sin aspectos definidos</th> </tr> </tbody> </table> <div style="margin-top: 10px;">    </div> </div> <div data-bbox="1013 1031 1382 1283" style="width: 35%;">  </div> </div>		Tarea: Dar Clases		Evaluación:	ADECUADO	Fecha:	2009-05-30 00:00:00	Observaciones de la tarea		No tiene observaciones		Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO		Tarea sin aspectos definidos	
Tarea: Dar Clases															
Evaluación:	ADECUADO														
Fecha:	2009-05-30 00:00:00														
Observaciones de la tarea															
No tiene observaciones															
Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO															
Tarea sin aspectos definidos															
Escenario 4: Mostrar Tarea Evaluada															
Curso normal de los eventos															
Actor	Respuesta del sistema														
1. El usuario solicita evaluar un usuario	2. El sistema le muestra un listado con todas las tareas que se le han asignado a dicho usuario.														

Flujo alterno 2																									
	2.1. Si el usuario no tiene asignada ninguna tarea el sistema lo notifica.																								
Prototipo de Interfaz																									
<div style="text-align: center;">  <p><b>EDISTIO YOEL VERDECIA MARTINEZ</b>  <i>JEFE DE DEPARTEMENTO del DEPARTAMENTO DE PROGRAMACION</i></p> </div> <h3 style="margin-top: 20px;">Listado de tareas a evaluar</h3> <div style="margin-top: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse; background-color: #e6f2ff;"> <tr><td colspan="2" style="text-align: center;">Tarea: Dar Clases</td></tr> <tr><td>Evaluacion: ADECUADO</td><td>Fecha: 2009-05-30 00:00:00</td></tr> <tr><td colspan="2" style="text-align: center;">Observaciones de la tarea</td></tr> <tr><td colspan="2" style="text-align: center;">No tiene observaciones</td></tr> <tr><td colspan="2">Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO</td></tr> <tr><td colspan="2" style="text-align: center;">Tarea sin aspectos definidos</td></tr> </table> <div style="margin-top: 5px; text-align: center;">  </div> </div> <div style="margin-top: 20px;"> <table border="1" style="width: 100%; border-collapse: collapse; background-color: #e6f2ff;"> <tr><td colspan="2" style="text-align: center;">Tarea: Lider de proyecto</td></tr> <tr><td>Evaluacion: ADECUADO</td><td>Fecha: 2009-05-11 00:00:00</td></tr> <tr><td colspan="2" style="text-align: center;">Observaciones de la tarea</td></tr> <tr><td colspan="2" style="text-align: center;">No tiene observaciones</td></tr> <tr><td colspan="2">Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO</td></tr> <tr><td colspan="2" style="text-align: center;">Tarea sin aspectos definidos</td></tr> </table> <div style="margin-top: 5px; text-align: center;">  </div> </div>		Tarea: Dar Clases		Evaluacion: ADECUADO	Fecha: 2009-05-30 00:00:00	Observaciones de la tarea		No tiene observaciones		Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO		Tarea sin aspectos definidos		Tarea: Lider de proyecto		Evaluacion: ADECUADO	Fecha: 2009-05-11 00:00:00	Observaciones de la tarea		No tiene observaciones		Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO		Tarea sin aspectos definidos	
Tarea: Dar Clases																									
Evaluacion: ADECUADO	Fecha: 2009-05-30 00:00:00																								
Observaciones de la tarea																									
No tiene observaciones																									
Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO																									
Tarea sin aspectos definidos																									
Tarea: Lider de proyecto																									
Evaluacion: ADECUADO	Fecha: 2009-05-11 00:00:00																								
Observaciones de la tarea																									
No tiene observaciones																									
Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO																									
Tarea sin aspectos definidos																									

Descripción extendida del caso de uso “Gestionar tarea evaluada”

CU_9	Gestionar Certificado	
Actor	Evaluador	
Propósito	Realizar los certificados de cada usuario evaluado del sistema.	
Descripción	El caso de uso se inicia cuando un evaluador, sea definidor o no, decida realizar el certificado de evaluación de alguno de sus usuarios a evaluar.	
Prioridad	Crítica	
Escenario 1: Adicionar Certificado		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El usuario solicita crear certificado.	2. El sistema muestra un listado con todas las tareas del usuario seleccionado. 3. Se validan los datos. 4. Se guardan los datos y se notifica.	
Flujo alternativo 2		
	2.1. Si al menos una de las tareas no está evaluada, se notifica.	
Flujo alternativo 3		
	3.1. Si existen campos vacíos se notifica.	

Prototipo de Interfaz





## Evaluado



MARIANNY HERNANDEZ BATISTA

Vicedecano del Vicedecanato de producción

	Tarea	Tipo	Fecha	Evaluación	Creador	Certificado
	Control de clases	Control		No evaluada	FRANCISCO JAVIER HERNANDEZ CAO	
Evaluación:	Superior ▼					
Tipo de evaluación:	enero Desde: lunes 1 de junio de 2009 Hasta: martes 30 de junio de 2009					
Observaciones:	<div style="border: 1px solid gray; height: 40px;"></div>					
Observaciones generales:	<div style="border: 1px solid gray; height: 40px;"></div>					


Escenario 2: Modificar Certificado															
Curso normal de los eventos															
Actor	Respuesta del sistema														
1. El usuario solicita modificar certificado.	2. El sistema muestra un formulario con todos los campos modificables. 3. Se almacenan los datos y se notifica.														
Flujo alterno															
	2.1. Si se modifican los datos, el sistema lo notifica.														
Prototipo de Interfaz															
<div style="border: 1px solid #ccc; padding: 10px;"> <h3>Evaluado</h3> <div style="display: flex; align-items: center;">  <div> <p><b>MARIANNY HERNANDEZ BATISTA</b> <i>Vicedecano del Vicedecanato de producción</i></p> </div> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th></th> <th>Tarea</th> <th>Tipo</th> <th>Fecha</th> <th>Evaluación</th> <th>Creador</th> <th>Certificado</th> </tr> </thead> <tbody> <tr> <td></td> <td>Control de clases</td> <td>Control</td> <td></td> <td>No evaluada</td> <td>FRANCISCO JAVIER HERNANDEZ CAO</td> <td></td> </tr> </tbody> </table> <div style="margin-top: 10px;"> <p>Evaluación: <span style="border: 1px solid #ccc; padding: 2px;">Superior ▼</span> lunes 8 de junio de 2009 11:08 am</p> <p>Tipo de evaluación: enero Desde: lunes 1 de junio de 2009 Hasta: martes 30 de junio de 2009</p> <p>Observaciones: <div style="border: 1px solid #ccc; padding: 5px; min-height: 40px;">la observación es...</div></p> <p>Observaciones generales: <div style="border: 1px solid #ccc; padding: 5px; min-height: 40px;"></div></p> </div> <div style="display: flex; align-items: center; margin-top: 10px;">    </div> </div>			Tarea	Tipo	Fecha	Evaluación	Creador	Certificado		Control de clases	Control		No evaluada	FRANCISCO JAVIER HERNANDEZ CAO	
	Tarea	Tipo	Fecha	Evaluación	Creador	Certificado									
	Control de clases	Control		No evaluada	FRANCISCO JAVIER HERNANDEZ CAO										

Escenario 3: Eliminar Certificado											
Curso normal de los eventos											
Actor	Respuesta del sistema										
1. El usuario solicita eliminar certificado.	2. El sistema debe permitir eliminar el certificado seleccionado. 3. El sistema muestra un mensaje de confirmación antes de eliminar. 4. Se notifica al usuario que eliminó el certificado.										
Flujo alternativo 2											
	2.1. Si el usuario no es el evaluador definidor, solo puede eliminar el certificado que él haya creado.										
Prototipo de Interfaz											
<p><b>Evaluado</b></p> <div style="display: flex; align-items: center;">  <div style="margin-left: 10px;"> <p><b>MARIANNY HERNANDEZ BATISTA</b>  <i>Vicedecano del Vicedecanato de producción</i></p> </div> <div style="margin-left: 20px;">  </div> </div> <p><b>CERTIFICADO DE EVALUACIÓN DEFINIDOR</b></p> <table border="1"> <thead> <tr> <th>Evaluación</th> <th>Fecha</th> <th>Tipo de evaluación</th> <th>Creador</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>ADECUADO</td> <td>lunes 8 de junio de 2009</td> <td>enero</td> <td>FRANCISCO JAVIER HERNANDEZ CAO</td> <td>                       Detalles                 </td> </tr> </tbody> </table>		Evaluación	Fecha	Tipo de evaluación	Creador	Acciones	ADECUADO	lunes 8 de junio de 2009	enero	FRANCISCO JAVIER HERNANDEZ CAO	  Detalles
Evaluación	Fecha	Tipo de evaluación	Creador	Acciones							
ADECUADO	lunes 8 de junio de 2009	enero	FRANCISCO JAVIER HERNANDEZ CAO	  Detalles							


Escenario 4: Mostrar Certificado											
Curso normal de los eventos											
Actor	Respuesta del sistema										
1. El actor solicita la opción realizar certificado.	2. El sistema muestra un formulario con todos los certificados creados a un usuario.										
Flujo alterno											
	2.1. Si no existen certificados creados, el sistema lo notifica.										
Prototipo de Interfaz											
<p><b>Evaluado</b></p>  <p><b>MARIANNY HERNANDEZ BATISTA</b> Vicedecano del Vicedecanato de producción</p> <p><b>CERTIFICADO DE EVALUACIÓN DEFINIDOR</b></p> <table border="1"> <thead> <tr> <th>Evaluación</th> <th>Fecha</th> <th>Tipo de evaluación</th> <th>Creador</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>ADECUADO</td> <td>lunes 8 de junio de 2009</td> <td>enero</td> <td>FRANCISCO JAVIER HERNANDEZ CAO</td> <td>                   Detalles             </td> </tr> </tbody> </table>		Evaluación	Fecha	Tipo de evaluación	Creador	Acciones	ADECUADO	lunes 8 de junio de 2009	enero	FRANCISCO JAVIER HERNANDEZ CAO	  Detalles
Evaluación	Fecha	Tipo de evaluación	Creador	Acciones							
ADECUADO	lunes 8 de junio de 2009	enero	FRANCISCO JAVIER HERNANDEZ CAO	  Detalles							


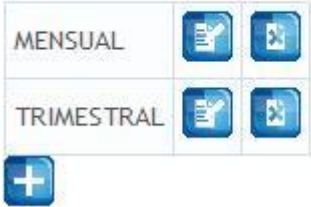
Descripción extendida del caso de uso “Gestionar certificado”

CU_10	Gestionar Tipo de Evaluación
Actor	Administrador
Propósito	Permitir al administrador gestionar los tipos de evaluación que van a existir en el sistema. <ul style="list-style-type: none"> <li>• Mensual</li> <li>• Trimestral</li> </ul>


Descripción	El caso de uso se inicia cuando el administrador solicita insertar un tipo de evaluación, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.	
Prioridad	Crítica	
Escenario 1: Adicionar Tipo de Evaluación		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción adicionar.	2. Se muestra un formulario a llenar y se le da la posibilidad al administrador de crear un tipo de evaluación. 3. Se validan los datos entrados. 4. Se almacena el tipo de evaluación adicionado. 5. Se notifica que se creó el tipo de evaluación y se finaliza en caso de uso.	
Flujo alterno 2		
	2.1. Si el tipo de evaluación ya existe el sistema lo notifica y da la posibilidad de crear uno nuevo.	
Prototipo de Interfaz		
		

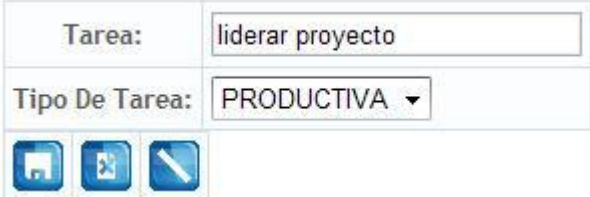



Escenario 2: Modificar Tipo de Evaluación	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El administrador selecciona la opción modificar.	2. Se muestra un formulario con todos los tipos de evaluación creados. 3. Se muestra un formulario con todos los datos modificables. 4. Se almacenan los datos y finaliza el caso de uso.
Flujo alternativo 2	
	2.1. Si no existen tipos de evaluación en el sistema no se permiten cambios.
Flujo alternativo 3	
	3.1. Si se modifican los datos el sistema lo notifica.
Prototipo de Interfaz	
	
Escenario 3: Eliminar Tipo de Evaluación	
Curso normal de los eventos	
Actor	Respuesta del sistema
2. El actor selecciona la opción eliminar	3. El sistema permite eliminar el tipo de evaluación seleccionado. 4. Se notifica que el tipo de evaluación se eliminó satisfactoriamente.

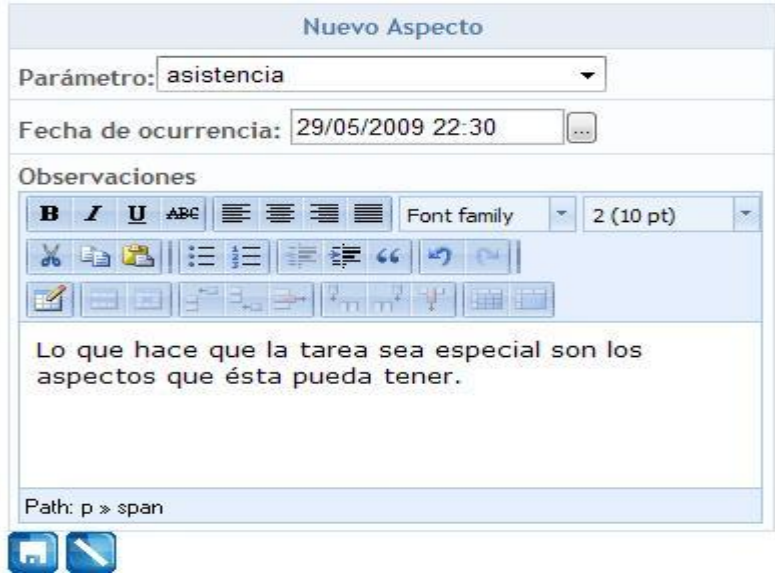
Flujo alternativo 2	
	2.1. Se muestra un mensaje de confirmación antes de eliminar definitivamente.
Prototipo de Interfaz	
	
Escenario 4: Mostrar Tipo de Evaluación	
Curso normal de los eventos	
Actor	Respuesta del sistema
5. El usuario solicita gestionar tipo de evaluación.	6. El sistema le muestra un listado con los tipos de evaluación existentes.
Flujo alternativo 2	
	2.1. Si el listado esta vacío, el sistema lo notifica.
Prototipo de Interfaz	
	

Descripción extendida del caso de uso “Gestionar tipo evaluación”

CU_11	Gestionar Tarea	
Actor	Administrador	
Propósito	Permitir al administrador gestionar las tareas que se les asignarán a los usuarios del sistema.	
Descripción	El caso de uso se inicia cuando el administrador solicita insertar una tarea, el sistema le muestra las funcionalidades de gestión y él selecciona la que se corresponda con la acción que vaya a realizar.	
Prioridad	Crítica	
Escenario 1: Adicionar Tarea		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción adicionar.	2. Se muestra un formulario a llenar y se le da la posibilidad al administrador de crear una tarea. 3. Se validan los datos entrados. 4. Se almacena la tarea adicionada 5. Se notifica que se creó la tarea y se finaliza en caso de uso.	
Flujo alternativo 2		
	2.1. Si la tarea ya existe el sistema lo notifica y da la posibilidad de crear una nueva.	
Prototipo de Interfaz		
		
Escenario 2: Modificar Tarea		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción modificar.	2. Se muestra un formulario con todas las tareas creadas. 3. Se muestra un formulario con todos los datos modificables. 4. Se almacenan los datos y finaliza el caso de uso.	

Flujo alternativo 3																	
	2.1. Si se modifican los datos el sistema lo notifica.																
Prototipo de Interfaz																	
																	
Escenario 3: Eliminar Tarea																	
Curso normal de los eventos																	
Actor	Respuesta del sistema																
1. El actor selecciona la opción eliminar	2. El sistema permite eliminar la tarea seleccionada. 3. Se notifica que la tarea se eliminó satisfactoriamente.																
Flujo alternativo 2																	
	2.1. Se muestra un mensaje de confirmación antes de eliminar definitivamente.																
Prototipo de Interfaz																	
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p><b>Listado de tareas</b></p> <table border="1"> <thead> <tr> <th>Tarea</th> <th>Tipo de tarea</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>Dar Clases</td> <td>DOCENTE</td> <td></td> <td></td> </tr> <tr> <td>Lider de proyecto</td> <td>PRODUCTIVA</td> <td></td> <td></td> </tr> <tr> <td>guardia obrera</td> <td>otro tipo</td> <td></td> <td></td> </tr> </tbody> </table> <p></p> </div> <div style="width: 45%;">  </div> </div>		Tarea	Tipo de tarea			Dar Clases	DOCENTE			Lider de proyecto	PRODUCTIVA			guardia obrera	otro tipo		
Tarea	Tipo de tarea																
Dar Clases	DOCENTE																
Lider de proyecto	PRODUCTIVA																
guardia obrera	otro tipo																

Descripción extendida del caso de uso “Gestionar tarea”

CU_12	Gestionar Tarea Especial	
Actor	Evaluador	
Propósito	Permitir al administrador gestionar las tareas especiales que se les asignarán a los usuarios del sistema.	
Descripción	El caso de uso se inicia cuando el administrador solicita agregarle aspectos a las tareas e evaluar del usuario seleccionado, el sistema le muestra todos los aspectos que puede agregarle a una tarea dada.	
Prioridad	Crítica	
Escenario 1: Adicionar Tarea Especial		
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción adicionar.	2. Se muestra un formulario con las tareas evaluadas y los parámetros creados y se le da la posibilidad al administrador de crear una tarea especial. 3. Se validan los datos entrados. 4. Se almacena la tarea adicionada 5. Se notifica que se creó la tarea especial.	
Prototipo de Interfaz		
		

Escenario 2: Modificar Tarea Especial									
Curso normal de los eventos									
Actor	Respuesta del sistema								
1. El administrador selecciona la opción modificar.	2. Se muestra un formulario con todas las tareas especiales creadas. 3. Se muestra un formulario con todos los datos modificables. 4. Se almacenan los datos.								
Flujo alterno 3									
	2.2. Si se modifican los datos el sistema lo notifica.								
Prototipo de Interfaz									
<div style="text-align: center;">  <p><b>EDISTIO YOEL VERDECIA MARTINEZ</b>  <i>JEFE DE DEPARTAMENTO del DEPARTAMENTO DE PROGRAMACION</i></p> </div> <div style="margin-top: 20px;"> <p style="text-align: center;"><b>Tarea: Dar Clases</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Evaluada por:</td> <td>FRANCISCO JAVIER HERNANDEZ CAO</td> </tr> <tr> <td>Evaluacion:</td> <td>ADECUADO ▾</td> </tr> <tr> <td>Fecha de evaluacion:</td> <td>30/05/2009 00:00 ...</td> </tr> <tr> <td>¿ Es tarea extra?</td> <td><input type="checkbox"/></td> </tr> </table> <div style="margin-top: 10px;"> <p><b>Observaciones</b></p> <div style="border: 1px solid #ccc; padding: 5px;"> <div style="font-size: 0.8em; margin-bottom: 5px;"> <span><b>B</b></span> <span><i>I</i></span> <span><u>U</u></span> <span>ABC</span> <span style="margin-left: 10px;">☰</span> <span style="margin-left: 10px;">☰</span> <span style="margin-left: 10px;">☰</span> <span style="margin-left: 10px;">☰</span> <span style="margin-left: 10px;">Font family ▾</span> <span style="margin-left: 10px;">Font size ▾</span> </div> <div style="font-size: 0.7em; margin-bottom: 5px;"> <span>✂</span> <span>📄</span> <span>📁</span> <span style="margin-left: 10px;">☰</span> <span style="margin-left: 10px;">☰</span> <span style="margin-left: 10px;">☰</span> <span style="margin-left: 10px;">☰</span> <span style="margin-left: 10px;">↶</span> <span style="margin-left: 10px;">↷</span> </div> <div style="font-size: 0.7em; margin-bottom: 5px;"> <span>📝</span> <span>📧</span> <span>📧</span> <span style="margin-left: 10px;">🔍</span> <span style="margin-left: 10px;">🔍</span> <span style="margin-left: 10px;">🔍</span> <span style="margin-left: 10px;">🔍</span> <span style="margin-left: 10px;">🔍</span> <span style="margin-left: 10px;">🔍</span> </div> <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> <p>dsfs</p> </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;">                     Path:                 </div> </div> </div> <div style="margin-top: 10px; text-align: center;">  </div> </div>		Evaluada por:	FRANCISCO JAVIER HERNANDEZ CAO	Evaluacion:	ADECUADO ▾	Fecha de evaluacion:	30/05/2009 00:00 ...	¿ Es tarea extra?	<input type="checkbox"/>
Evaluada por:	FRANCISCO JAVIER HERNANDEZ CAO								
Evaluacion:	ADECUADO ▾								
Fecha de evaluacion:	30/05/2009 00:00 ...								
¿ Es tarea extra?	<input type="checkbox"/>								


Escenario 3: Eliminar Tarea Especial																													
Curso normal de los eventos																													
Actor	Respuesta del sistema																												
1. El actor selecciona la opción eliminar	2. El sistema permite eliminar la tarea especial seleccionada. 3. Se notifica que la tarea especial se eliminó satisfactoriamente.																												
Flujo alternativo 2																													
	2.1. Se muestra un mensaje de confirmación antes de eliminar definitivamente.																												
Prototipo de Interfaz																													
 <p><b>EDISTIO YOEL VERDECIA MARTINEZ</b> JEFE DE DEPARTEMENTO del DEPARTAMENTO DE PROGRAMACION</p> <h3>Listado de tareas a evaluar</h3> <table border="1"> <thead> <tr> <th colspan="4">Tarea: Dar Clases</th> </tr> <tr> <td>Evaluacion: ADECUADO</td> <td></td> <td>Fecha: 2009</td> <td></td> </tr> <tr> <th colspan="4">Observaciones de la tarea</th> </tr> <tr> <td colspan="4">No tiene observaciones</td> </tr> <tr> <td colspan="4">Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO</td> </tr> <tr> <th>Aspectos</th> <th>Fecha de ocurrencia</th> <th>Observaciones</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>asistencia</td> <td>2009-05-29 22:30:00</td> <td>Lo que hace que la tarea sea especial son los aspectos que ésta pueda tener.</td> <td> </td> </tr> </tbody> </table> <p>  </p>		Tarea: Dar Clases				Evaluacion: ADECUADO		Fecha: 2009		Observaciones de la tarea				No tiene observaciones				Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO				Aspectos	Fecha de ocurrencia	Observaciones	Acciones	asistencia	2009-05-29 22:30:00	Lo que hace que la tarea sea especial son los aspectos que ésta pueda tener.	 
Tarea: Dar Clases																													
Evaluacion: ADECUADO		Fecha: 2009																											
Observaciones de la tarea																													
No tiene observaciones																													
Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO																													
Aspectos	Fecha de ocurrencia	Observaciones	Acciones																										
asistencia	2009-05-29 22:30:00	Lo que hace que la tarea sea especial son los aspectos que ésta pueda tener.	 																										

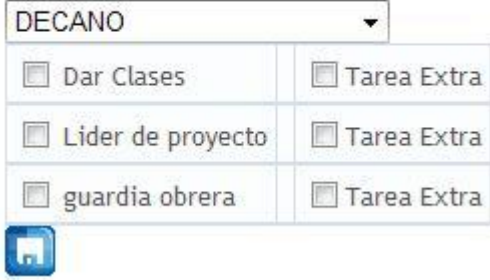



Escenario 4: Mostrar Tarea Especial																													
Curso normal de los eventos																													
Actor	Respuesta del sistema																												
1. El usuario solicita evaluar las tareas del usuario seleccionado.	2. El sistema le muestra un listado con todas las tareas que estén asignadas al mismo. Las tareas especiales son las que tienen aspectos que las distinguen.																												
Flujo alternativo 2																													
	2.1. Si no existen tareas evaluadas el sistema lo notifica.																												
Prototipo de Interfaz																													
<div style="display: flex; align-items: center;">  <div> <p><b>EDISTIO YOEL VERDECIA MARTINEZ</b>  <i>JEFE DE DEPARTEMENTO del DEPARTAMENTO DE PROGRAMACION</i></p> </div> </div> <p><b>Listado de tareas a evaluar</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4" style="background-color: #e1eef6;">Tarea: Dar Clases</th> </tr> </thead> <tbody> <tr> <td style="width: 25%;">Evaluación: ADECUADO</td> <td colspan="3">Fecha: 2009-05-30 00:00:00</td> </tr> <tr> <th colspan="4" style="background-color: #e1eef6;">Observaciones de la tarea</th> </tr> <tr> <td colspan="4" style="text-align: center;">No tiene observaciones</td> </tr> <tr> <td colspan="4" style="text-align: center;">Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO</td> </tr> <tr> <th style="width: 25%;">Aspectos</th> <th style="width: 20%;">Fecha de ocurrencia</th> <th style="width: 45%;">Observaciones</th> <th style="width: 10%;">Acciones</th> </tr> <tr> <td>asistencia</td> <td>2009-05-29 22:30:00</td> <td>Lo que hace que la tarea sea especial son los aspectos que ésta pueda tener...</td> <td style="text-align: center;">   </td> </tr> </tbody> </table> <div style="margin-top: 10px;">    </div>		Tarea: Dar Clases				Evaluación: ADECUADO	Fecha: 2009-05-30 00:00:00			Observaciones de la tarea				No tiene observaciones				Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO				Aspectos	Fecha de ocurrencia	Observaciones	Acciones	asistencia	2009-05-29 22:30:00	Lo que hace que la tarea sea especial son los aspectos que ésta pueda tener...	 
Tarea: Dar Clases																													
Evaluación: ADECUADO	Fecha: 2009-05-30 00:00:00																												
Observaciones de la tarea																													
No tiene observaciones																													
Tarea adicionada por FRANCISCO JAVIER HERNANDEZ CAO																													
Aspectos	Fecha de ocurrencia	Observaciones	Acciones																										
asistencia	2009-05-29 22:30:00	Lo que hace que la tarea sea especial son los aspectos que ésta pueda tener...	 																										

Descripción extendida del caso de uso “Gestionar tarea especial”




CU_13	Asignar Tareas							
Actor	Administrador, Evaluador							
Propósito	Permitir al administrador asignarle a las áreas, los cargos, los cargos de un área en específico y los usuarios las tareas generales para cada uno respectivamente							
Descripción	El caso de uso se inicia cuando el administrador solicita asignar una tarea, así sean tareas generales para un cargo o un área, un cargo de un área dada, o para un usuario en específico.							
Prioridad	Crítica							
Escenario 1: Asignar tareas a área								
Curso normal de los eventos								
Actor	Respuesta del sistema							
1. El administrador solicita insertar tareas a las áreas del sistema.	2. El sistema muestra las asignaciones existentes. 3. Se permite que el usuario seleccione las tareas que desea asignar a un área determinada. 4. Se guardan los datos. 5. Se notifica que se guardaron los datos.							
Flujo alternativo 3								
	3.1. Si no existen tareas o áreas, el sistema lo notifica.							
Prototipo de Interfaz								
<p><b>Tareas asignadas por área</b></p> <p>VICEDECANATO DE PRODUCCION ▼</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><input type="checkbox"/> Dar Clases</td> <td style="text-align: center;"><input type="checkbox"/> Tarea Extra</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/> Lider de proyecto</td> <td style="text-align: center;"><input type="checkbox"/> Tarea Extra</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/> guardia obrera</td> <td style="text-align: center;"><input type="checkbox"/> Tarea Extra</td> </tr> </table> <p style="text-align: center;"></p>			<input type="checkbox"/> Dar Clases	<input type="checkbox"/> Tarea Extra	<input type="checkbox"/> Lider de proyecto	<input type="checkbox"/> Tarea Extra	<input type="checkbox"/> guardia obrera	<input type="checkbox"/> Tarea Extra
<input type="checkbox"/> Dar Clases	<input type="checkbox"/> Tarea Extra							
<input type="checkbox"/> Lider de proyecto	<input type="checkbox"/> Tarea Extra							
<input type="checkbox"/> guardia obrera	<input type="checkbox"/> Tarea Extra							

Escenario 2: Asignar tareas por cargo	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El administrador selecciona la opción de asignar tareas a los cargos.	2. El sistema muestra las asignaciones existentes. 3. Se permite que el usuario seleccione las tareas que desea asignar a un cargo determinado. 4. Se guardan los datos. 5. Se notifica que se guardaron los datos.
Flujo alternativo 3	
	3.2. Si no existen tareas o cargos, el sistema lo notifica.
Prototipo de Interfaz	
<p><b>Asignar tareas por cargo</b></p> 	
Escenario 3: Asignar tareas a cargos de un área específica	
Curso normal de los eventos	
Actor	Respuesta del sistema
1. El administrador solicita asignar tareas a un cargo de un área específica.	2. El sistema muestra las asignaciones existentes. 3. Se permite que el usuario seleccione las tareas que desea asignar a un cargo determinado. 4. Se guardan los datos. 5. Se notifica que se guardaron los datos.
Flujo alternativo 3	
	3.1. Si no existen tareas o cargos asignados a las áreas, el sistema lo notifica.

Prototipo de Interfaz							
<p><b>Asignar tareas para un cargo de un área determinada</b></p> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 5px;"> <span>DECANO del DECANATO ▼</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><input type="checkbox"/> Dar Clases</td> <td style="padding: 2px;"><input type="checkbox"/> Tarea Extra</td> </tr> <tr> <td style="padding: 2px;"><input type="checkbox"/> Lider de proyecto</td> <td style="padding: 2px;"><input type="checkbox"/> Tarea Extra</td> </tr> <tr> <td style="padding: 2px;"><input type="checkbox"/> guardia obrera</td> <td style="padding: 2px;"><input type="checkbox"/> Tarea Extra</td> </tr> </table> 		<input type="checkbox"/> Dar Clases	<input type="checkbox"/> Tarea Extra	<input type="checkbox"/> Lider de proyecto	<input type="checkbox"/> Tarea Extra	<input type="checkbox"/> guardia obrera	<input type="checkbox"/> Tarea Extra
<input type="checkbox"/> Dar Clases	<input type="checkbox"/> Tarea Extra						
<input type="checkbox"/> Lider de proyecto	<input type="checkbox"/> Tarea Extra						
<input type="checkbox"/> guardia obrera	<input type="checkbox"/> Tarea Extra						
Escenario 4: Asignar tareas a usuario							
Curso normal de los eventos							
Actor	Respuesta del sistema						
<p>1. El evaluador solicita asignar tareas a un usuario.</p>	<p>2. El sistema muestra las asignaciones existentes.                      3. Se permite que el usuario seleccione las tareas que desea asignar a un usuario.                      4. Se guardan los datos.                      5. Se notifica que se guardaron los datos.</p>						
Flujo alternativo 3							
	<p>3.1. Si no existen tareas o usuarios, el sistema lo notifica.</p>						

Prototipo de Interfaz



**EDISTIO YOEL VERDECIA MARTINEZ**  
*JEFE DE DEPARTAMENTO del DEPARTAMENTO DE PROGRAMACION*

Nueva Tarea:	<input type="text"/>
Tipo de tarea:	DOCENTE ▾
Evaluacion:	SUPERIOR ▾
Fecha de evaluacion:	<input type="text"/> ...
¿ Es tarea extra?	<input type="checkbox"/>



Observaciones

**B** *I* **U** AB ≡ ≡ ≡ ≡ Font family ▾ Font size ▾

✂ 📄 📁 ☰ ☰ ☰ ☰ ☰ ☰ ☰

📝 📄 📁 ☰ ☰ ☰ ☰ ☰ ☰

Path:





Descripción extendida del caso de uso “Gestionar asignación de tareas”


CU_14	Asignar funcionalidades																						
Actor	Administrador																						
Propósito	Permitir al administrador asignarle funcionalidades a los cargos.																						
Descripción	El caso de uso se inicia cuando el administrador solicita asignarle las funcionalidades necesarias a los cargos que tengan los usuarios del sistema.																						
Prioridad	Crítica																						
Curso normal de los eventos																							
Actor	Respuesta del sistema																						
1. El usuario selecciona asignar tarea a usuario.	2. El sistema debe mostrar las asignaciones existentes. 3. Se guardan los datos. 4. Se notifica que se guardaron los datos.																						
Flujo alternativo 2																							
	2.1. Si no existen cargos, el sistema lo notifica.																						
Prototipo de Interfaz																							
<div style="text-align: center;"> <h2>Webmaster</h2> <p><b>Funcionalidades:</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><input type="checkbox"/> Ver Areas</td> <td><input type="checkbox"/> Crear Area</td> <td><input type="checkbox"/> Eliminar Area</td> <td><input type="checkbox"/> Editar Area</td> </tr> <tr> <td><input type="checkbox"/> Ver Certificado</td> <td><input type="checkbox"/> Crear Certificado</td> <td><input type="checkbox"/> Editar Certificado</td> <td><input type="checkbox"/> Eliminar Certificado</td> </tr> <tr> <td><input type="checkbox"/> Ver Tarea</td> <td><input type="checkbox"/> Crear Tarea</td> <td><input type="checkbox"/> Editar Tareas</td> <td><input type="checkbox"/> Eliminar Tarea</td> </tr> <tr> <td><input type="checkbox"/> Ver Usuario</td> <td><input type="checkbox"/> Crear Usuario</td> <td><input type="checkbox"/> Editar Usuario</td> <td><input type="checkbox"/> Eliminar Usuario</td> </tr> <tr> <td><input type="checkbox"/> Ver Usuario Local</td> <td><input type="checkbox"/> Crear Usuario Local</td> <td><input type="checkbox"/> Editar Usuario Local</td> <td><input type="checkbox"/> Eliminar Usuario Local</td> </tr> </table> </div>				<input type="checkbox"/> Ver Areas	<input type="checkbox"/> Crear Area	<input type="checkbox"/> Eliminar Area	<input type="checkbox"/> Editar Area	<input type="checkbox"/> Ver Certificado	<input type="checkbox"/> Crear Certificado	<input type="checkbox"/> Editar Certificado	<input type="checkbox"/> Eliminar Certificado	<input type="checkbox"/> Ver Tarea	<input type="checkbox"/> Crear Tarea	<input type="checkbox"/> Editar Tareas	<input type="checkbox"/> Eliminar Tarea	<input type="checkbox"/> Ver Usuario	<input type="checkbox"/> Crear Usuario	<input type="checkbox"/> Editar Usuario	<input type="checkbox"/> Eliminar Usuario	<input type="checkbox"/> Ver Usuario Local	<input type="checkbox"/> Crear Usuario Local	<input type="checkbox"/> Editar Usuario Local	<input type="checkbox"/> Eliminar Usuario Local
<input type="checkbox"/> Ver Areas	<input type="checkbox"/> Crear Area	<input type="checkbox"/> Eliminar Area	<input type="checkbox"/> Editar Area																				
<input type="checkbox"/> Ver Certificado	<input type="checkbox"/> Crear Certificado	<input type="checkbox"/> Editar Certificado	<input type="checkbox"/> Eliminar Certificado																				
<input type="checkbox"/> Ver Tarea	<input type="checkbox"/> Crear Tarea	<input type="checkbox"/> Editar Tareas	<input type="checkbox"/> Eliminar Tarea																				
<input type="checkbox"/> Ver Usuario	<input type="checkbox"/> Crear Usuario	<input type="checkbox"/> Editar Usuario	<input type="checkbox"/> Eliminar Usuario																				
<input type="checkbox"/> Ver Usuario Local	<input type="checkbox"/> Crear Usuario Local	<input type="checkbox"/> Editar Usuario Local	<input type="checkbox"/> Eliminar Usuario Local																				

Descripción extendida del caso de uso "Gestionar asignación de funcionalidades"

CU_15	Asignar cargo																			
Actor	Administrador																			
Propósito	Permitir al administrador asignarle cargos a los usuarios y las áreas del sistema.																			
Descripción	El caso de uso se inicia cuando el administrador solicita asignarle cargos a los usuarios o las áreas del sistema.																			
Prioridad	Crítica																			
Curso normal de los eventos																				
Escenario1: Asignar cargos a usuario																				
Actor	Respuesta del sistema																			
1. El administrador selecciona la opción asignar.	2. Se muestra un listado de cargos y otro de usuarios, le da la posibilidad al administrador de asignarle cargos a un usuario. 3. Se almacena la asignación. 4. El sistema debe notificar que se asignaron los cargos y finaliza el caso de uso.																			
Flujo alterno 2																				
	2.1. Si no existen cargos o usuarios se debe mostrar un mensaje notificándolo.																			
Prototipo de Interfaz																				
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><b>Cargo Real:</b></td> <td><b>Solapin:</b> <input type="text" value="50074"/></td> <td><b>Es webmaster:</b> <input type="checkbox"/></td> </tr> <tr> <td><input type="radio"/></td> <td><input checked="" type="checkbox"/> Webmaster</td> <td>Ver áreas</td> </tr> <tr> <td><input checked="" type="radio"/></td> <td><input checked="" type="checkbox"/> Decano</td> <td>Ver áreas</td> </tr> <tr> <td><input type="radio"/></td> <td><input checked="" type="checkbox"/> Vicedecano</td> <td>Ver áreas</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/> Jefe de departamento</td> <td>Ver áreas</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/> Profesor</td> <td>Ver áreas</td> </tr> </table>			<b>Cargo Real:</b>	<b>Solapin:</b> <input type="text" value="50074"/>	<b>Es webmaster:</b> <input type="checkbox"/>	<input type="radio"/>	<input checked="" type="checkbox"/> Webmaster	Ver áreas	<input checked="" type="radio"/>	<input checked="" type="checkbox"/> Decano	Ver áreas	<input type="radio"/>	<input checked="" type="checkbox"/> Vicedecano	Ver áreas	<input type="radio"/>	<input type="checkbox"/> Jefe de departamento	Ver áreas	<input type="radio"/>	<input type="checkbox"/> Profesor	Ver áreas
<b>Cargo Real:</b>	<b>Solapin:</b> <input type="text" value="50074"/>	<b>Es webmaster:</b> <input type="checkbox"/>																		
<input type="radio"/>	<input checked="" type="checkbox"/> Webmaster	Ver áreas																		
<input checked="" type="radio"/>	<input checked="" type="checkbox"/> Decano	Ver áreas																		
<input type="radio"/>	<input checked="" type="checkbox"/> Vicedecano	Ver áreas																		
<input type="radio"/>	<input type="checkbox"/> Jefe de departamento	Ver áreas																		
<input type="radio"/>	<input type="checkbox"/> Profesor	Ver áreas																		


Curso normal de los eventos													
Escenario 2: Asignar cargos a áreas													
Actor	Respuesta del sistema												
5. El administrador selecciona la opción asignar.	6. Se muestra un listado de cargos y otro de usuarios, le da la posibilidad al administrador de asignarle cargos a un usuario. 7. Se almacena la asignación. 8. El sistema debe notificar que se asignaron los cargos y finaliza el caso de uso.												
Flujo alterno 2													
	2.2. Si no existen cargos o usuarios se debe mostrar un mensaje notificándolo.												
Prototipo de Interfaz													
<p><b>Asignación de cargos por áreas</b></p> <p><b>Áreas</b></p> <div style="border: 1px solid gray; padding: 5px; width: fit-content; margin: 0 auto;"> <input type="text" value="Sistema"/> </div> <table border="1" style="margin: 10px auto; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #e1eef6;">Cargos</th> <th></th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> Webmaster</td> <td></td> </tr> <tr> <td><input type="checkbox"/> Decano</td> <td></td> </tr> <tr> <td><input type="checkbox"/> Vicedecano</td> <td></td> </tr> <tr> <td><input type="checkbox"/> Jefe de departamento</td> <td></td> </tr> <tr> <td><input type="checkbox"/> Profesor</td> <td></td> </tr> </tbody> </table> 		Cargos		<input checked="" type="checkbox"/> Webmaster		<input type="checkbox"/> Decano		<input type="checkbox"/> Vicedecano		<input type="checkbox"/> Jefe de departamento		<input type="checkbox"/> Profesor	
Cargos													
<input checked="" type="checkbox"/> Webmaster													
<input type="checkbox"/> Decano													
<input type="checkbox"/> Vicedecano													
<input type="checkbox"/> Jefe de departamento													
<input type="checkbox"/> Profesor													

Descripción extendida del caso de uso "Asignar cargos"

CU_16	Asignar áreas	
Actor	Administrador	
Propósito	Permitir al administrador asignarles áreas a los usuarios del sistema.	
Descripción	El caso de uso se inicia cuando el administrador solicita asignarle el área a cada uno de los usuarios del sistema, es decir especificar a qué área va a pertenecer cada uno.	
Prioridad	Crítica	
Curso normal de los eventos		
Actor	Respuesta del sistema	
1. El administrador selecciona la opción asignar.	2. Se muestra un listado de áreas y otro de usuarios, le da la posibilidad al administrador de asignarle áreas a un usuario. 3. Se almacena la asignación. 4. Se notifica que se guardaron los cambios.	
Flujo alternativo 2		
	2.1. Si no existen áreas o usuarios se debe mostrar un mensaje notificándolo.	
Prototipo de Interfaz		
		

Descripción extendida del caso de uso “Gestionar asignación de áreas”



CU_18	Autenticar usuario	
Actor	<ul style="list-style-type: none"> <li>• Administrador</li> <li>• Evaluador</li> <li>• Evaluado</li> </ul>	
Propósito	Permitir a los usuarios autenticarse en el sistema.	
Descripción	El usuario solicita autenticarse, el sistema le muestra la pantalla de autenticación.	
Prioridad	Crítica	
Curso normal de los eventos		
Actor	Respuesta del sistema	
3. El usuario solicita autenticarse	4. El sistema muestra la pantalla de autenticación. 5. se muestran los campos: usuario y contraseña. 6. Se validan los datos. 7. El sistema muestra las opciones a las que el usuario tiene acceso.	
Flujo alternativo 4		
	1.1. Si el usuario no tiene permisos para acceder, el sistema lo notifica y le impide el acceso.	
Prototipo de Interfaz		
 <p>El prototipo de interfaz muestra un formulario con un título 'Autenticar' en un encabezado azul. Debajo del título, hay dos campos de entrada: 'Usuario' y 'Contraseña', cada uno con un cuadro de texto blanco y un borde gris. Debajo de los campos, hay un botón rectangular con el texto 'Entrar'.</p>		

Anexo 4: Diagramas de clases del diseño.

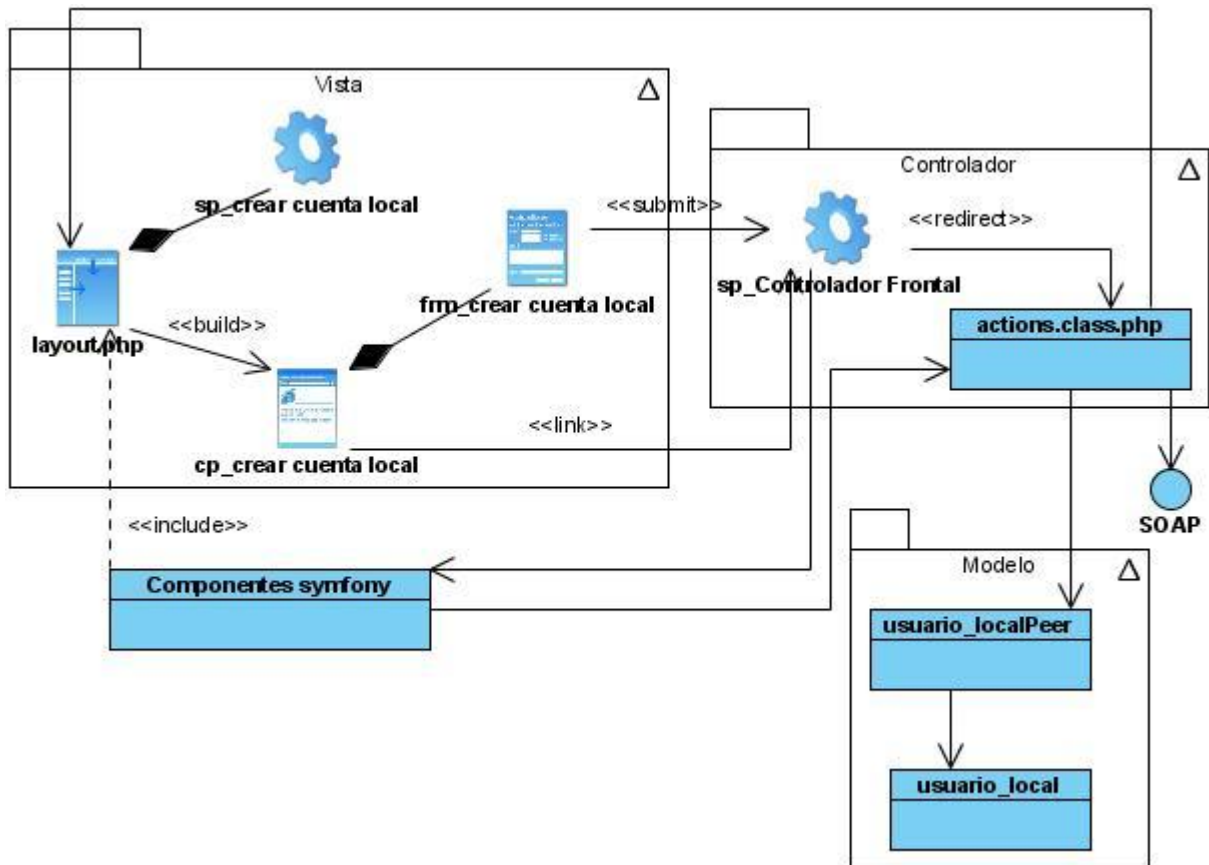


Diagrama de clases del diseño del caso de uso “Crear cuenta local”

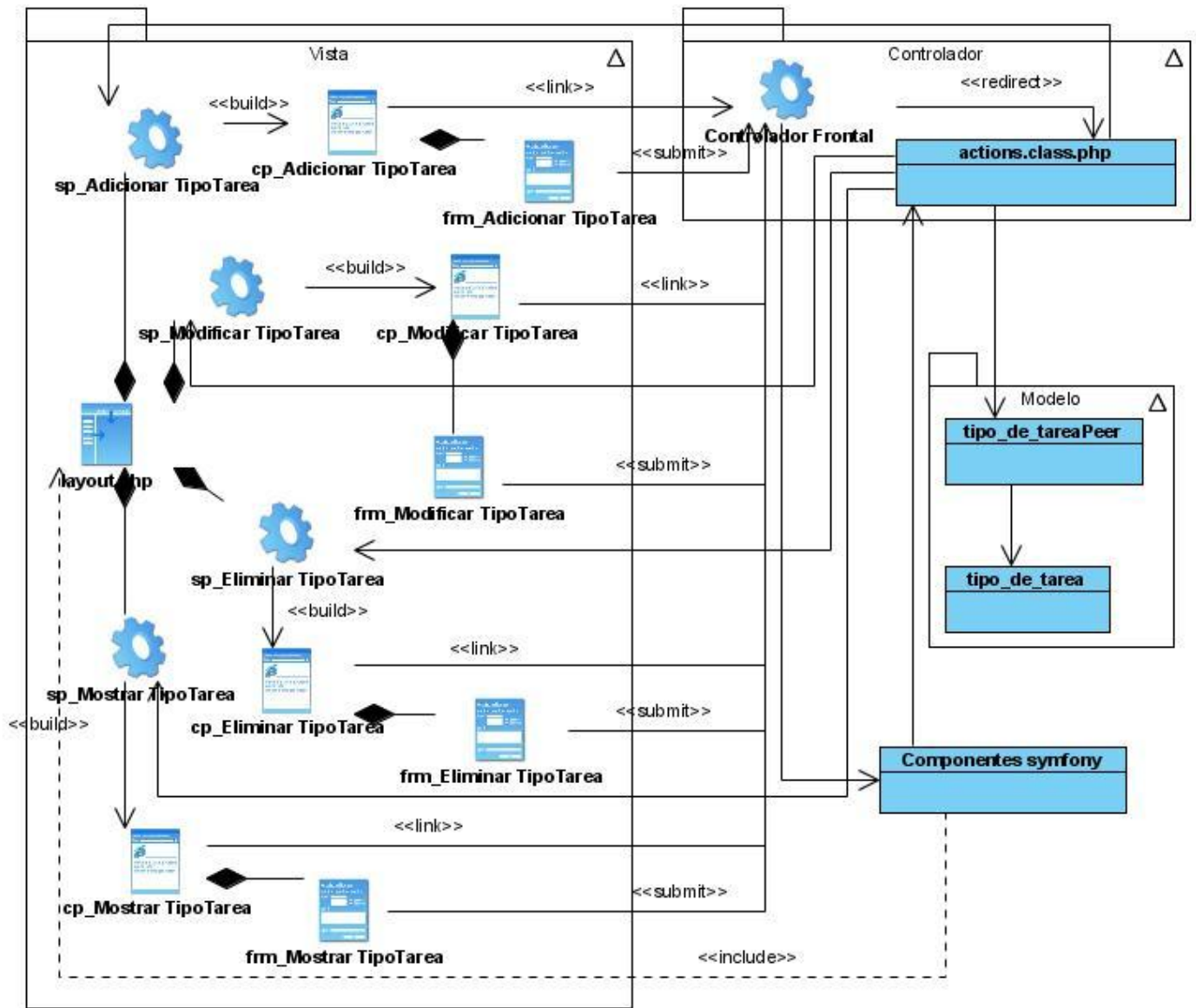


Diagrama de clases del diseño del caso de uso "Gestionar tipo de tarea"



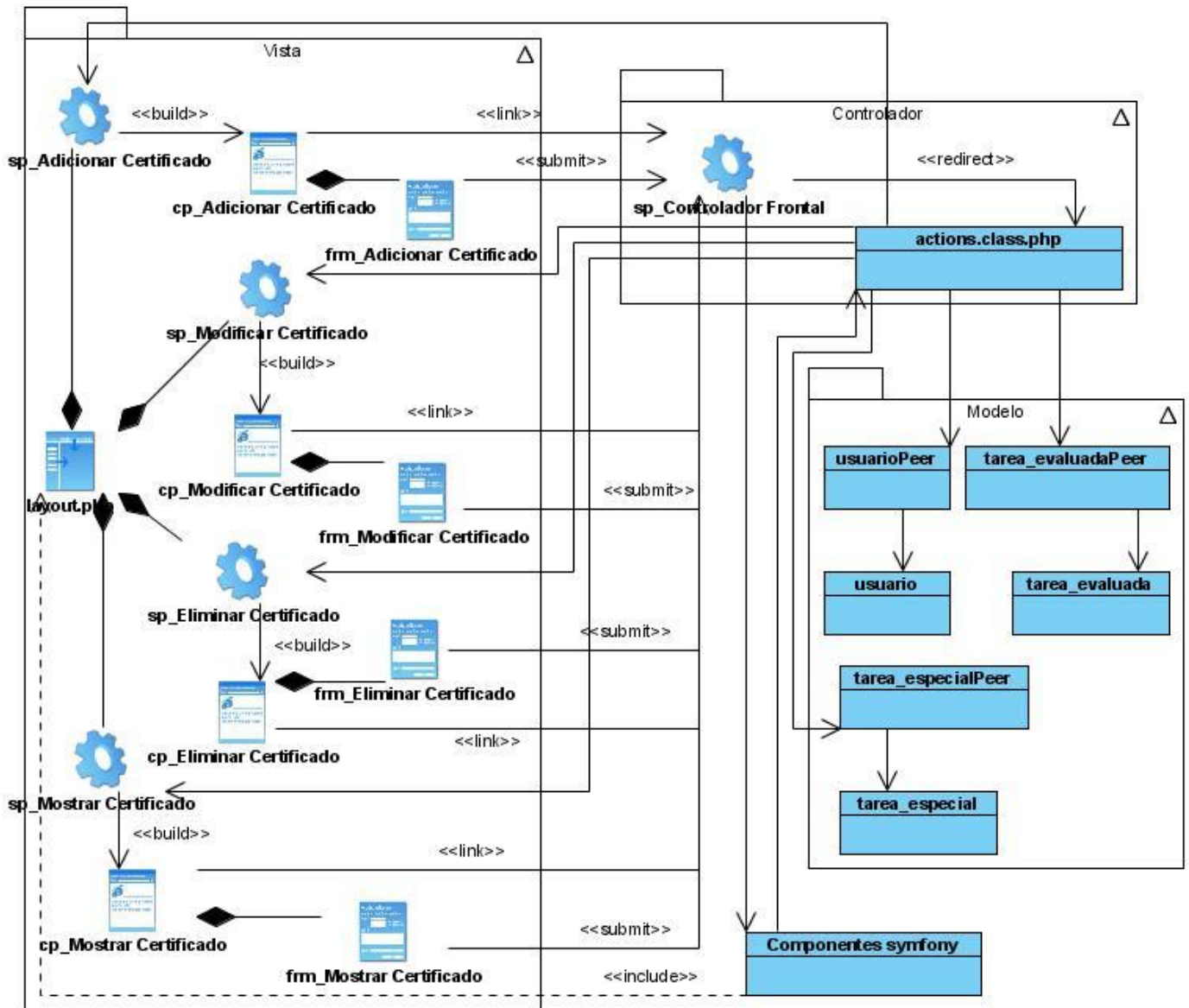


Diagrama de clases del diseño del caso de uso "Gestionar certificado"

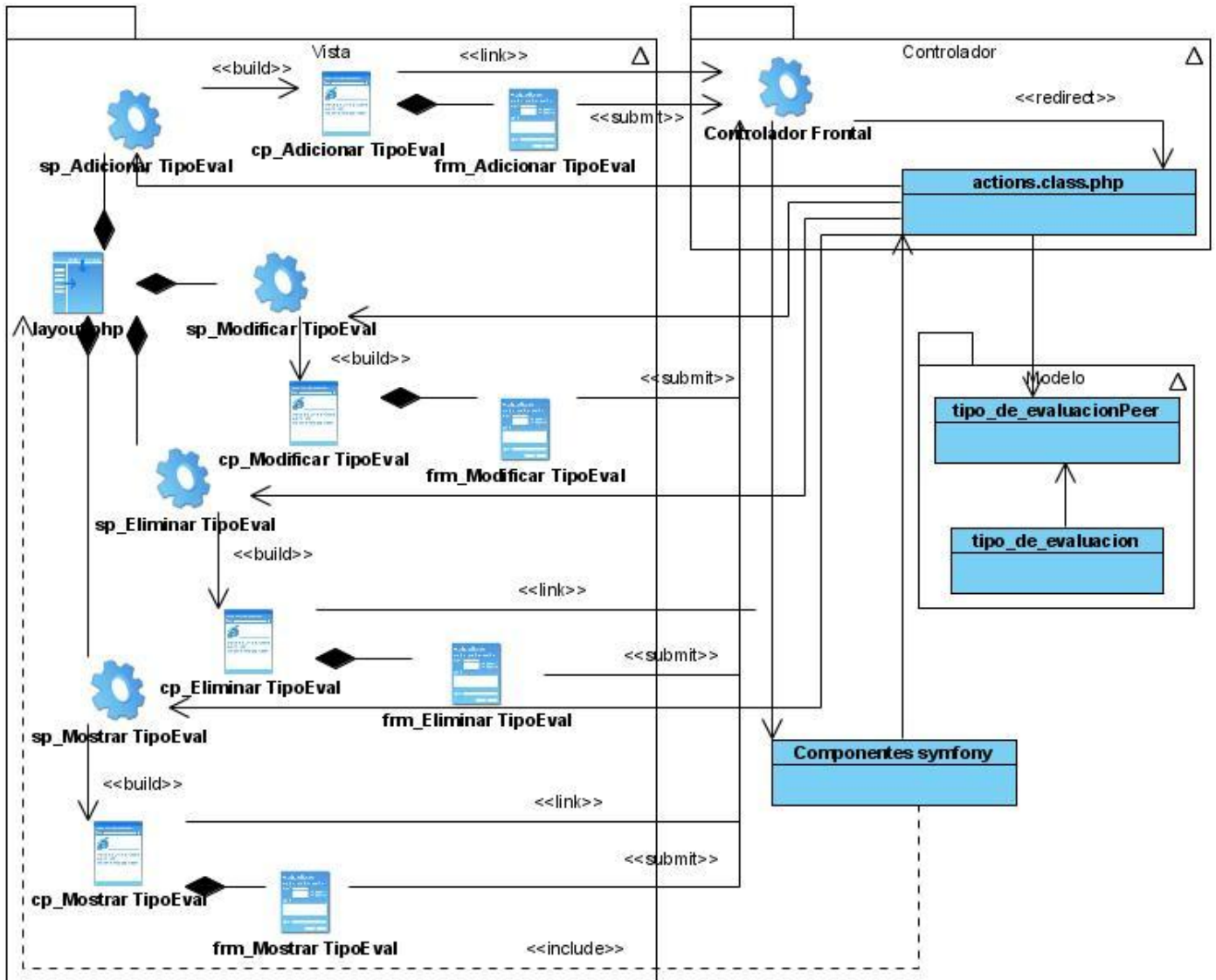


Diagrama de clases del diseño del caso de uso “Gestionar tipo de evaluación”

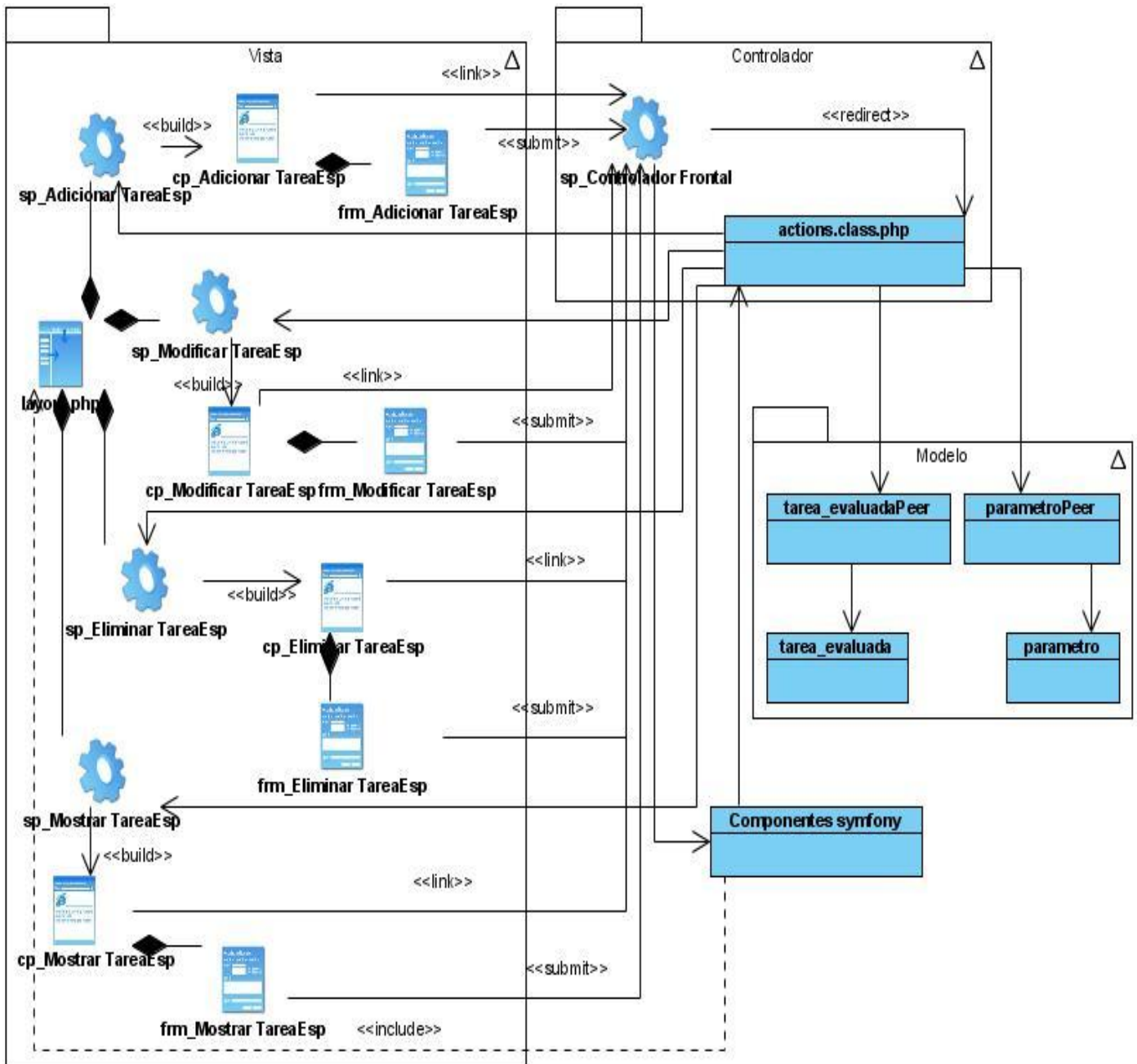


Diagrama de clases del diseño del caso de uso “Gestionar tarea especial”



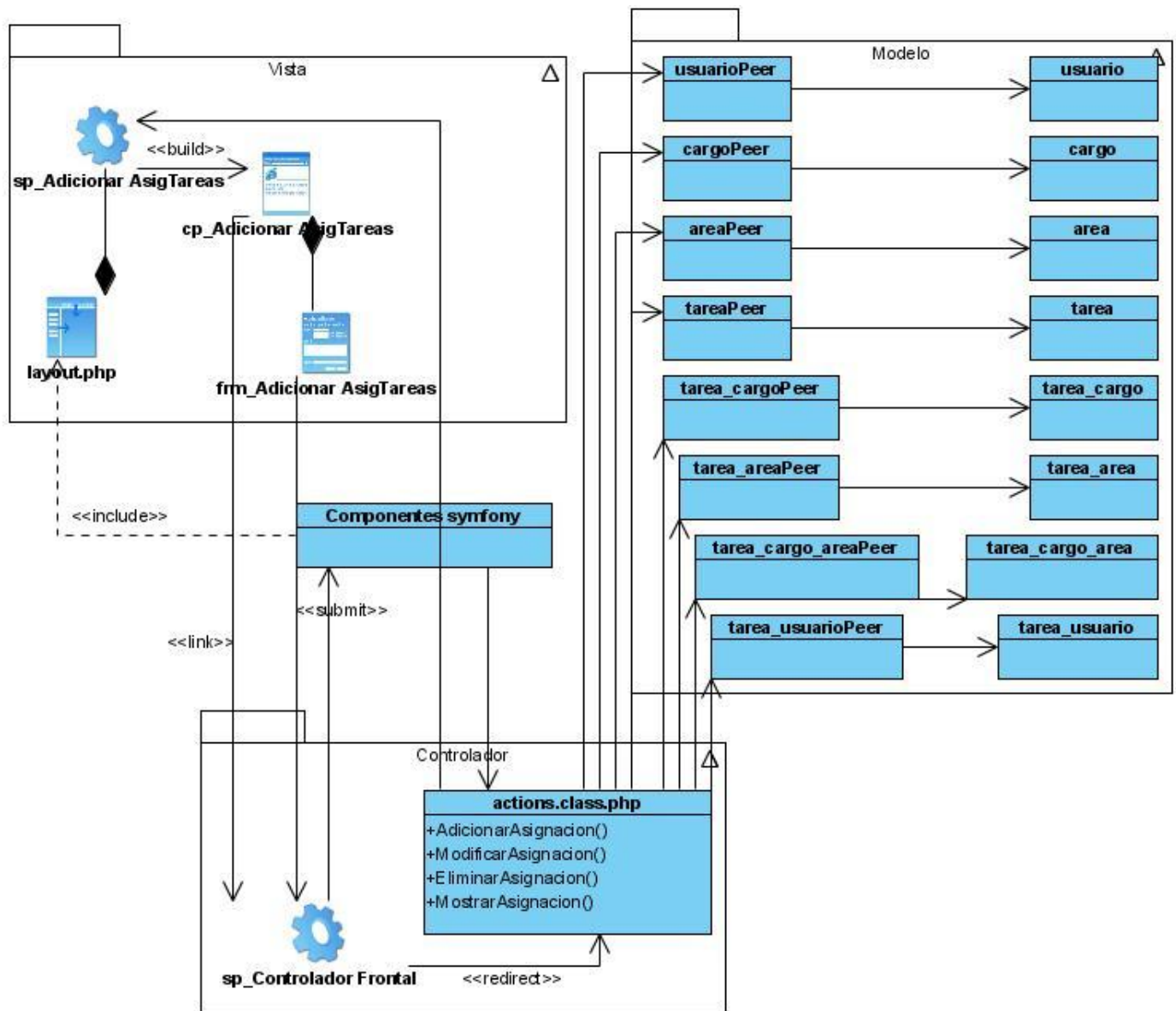


Diagrama de clases del diseño del caso de uso "Asignar tareas"



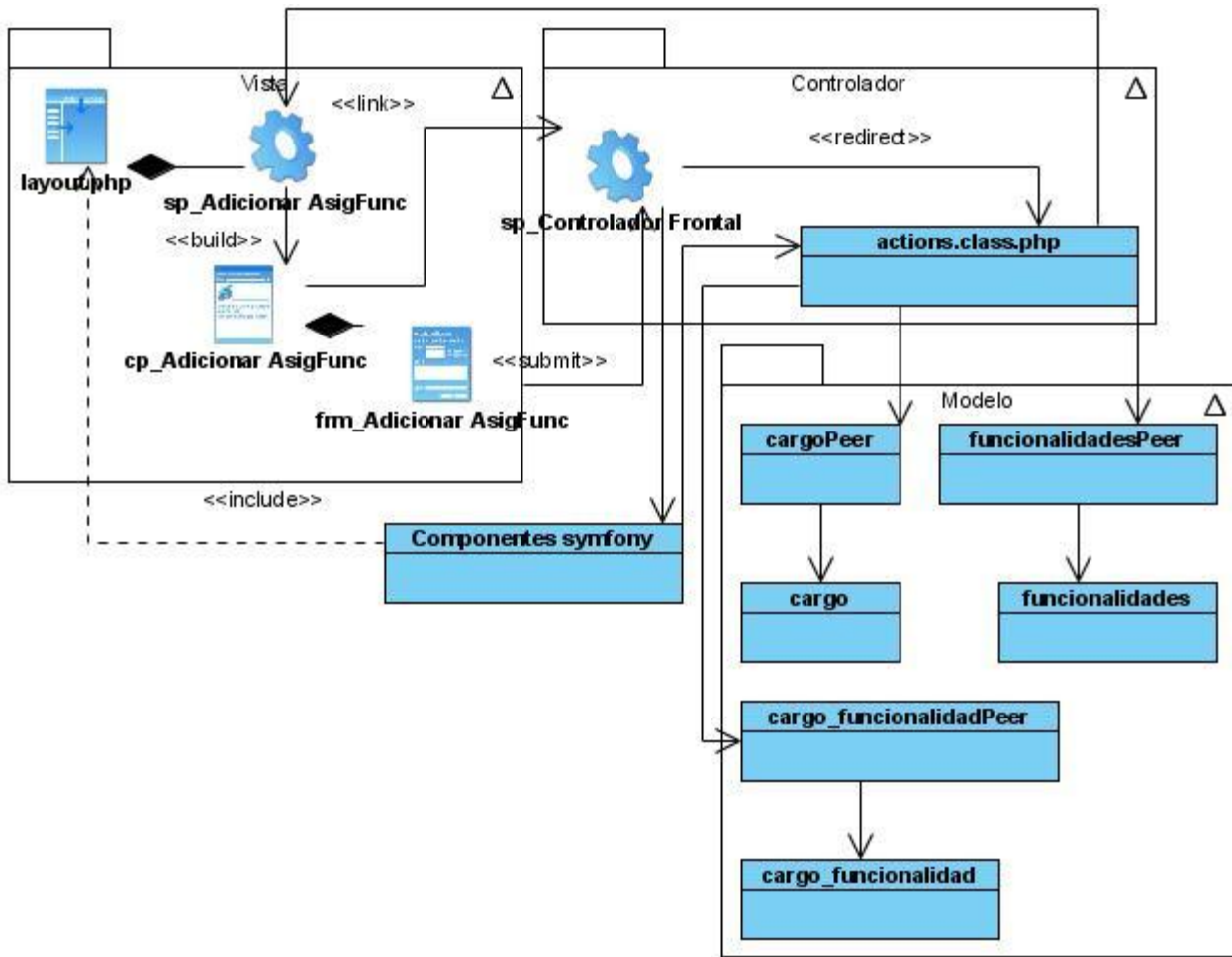


Diagrama de clases del diseño del caso de uso “Asignar funcionalidades”

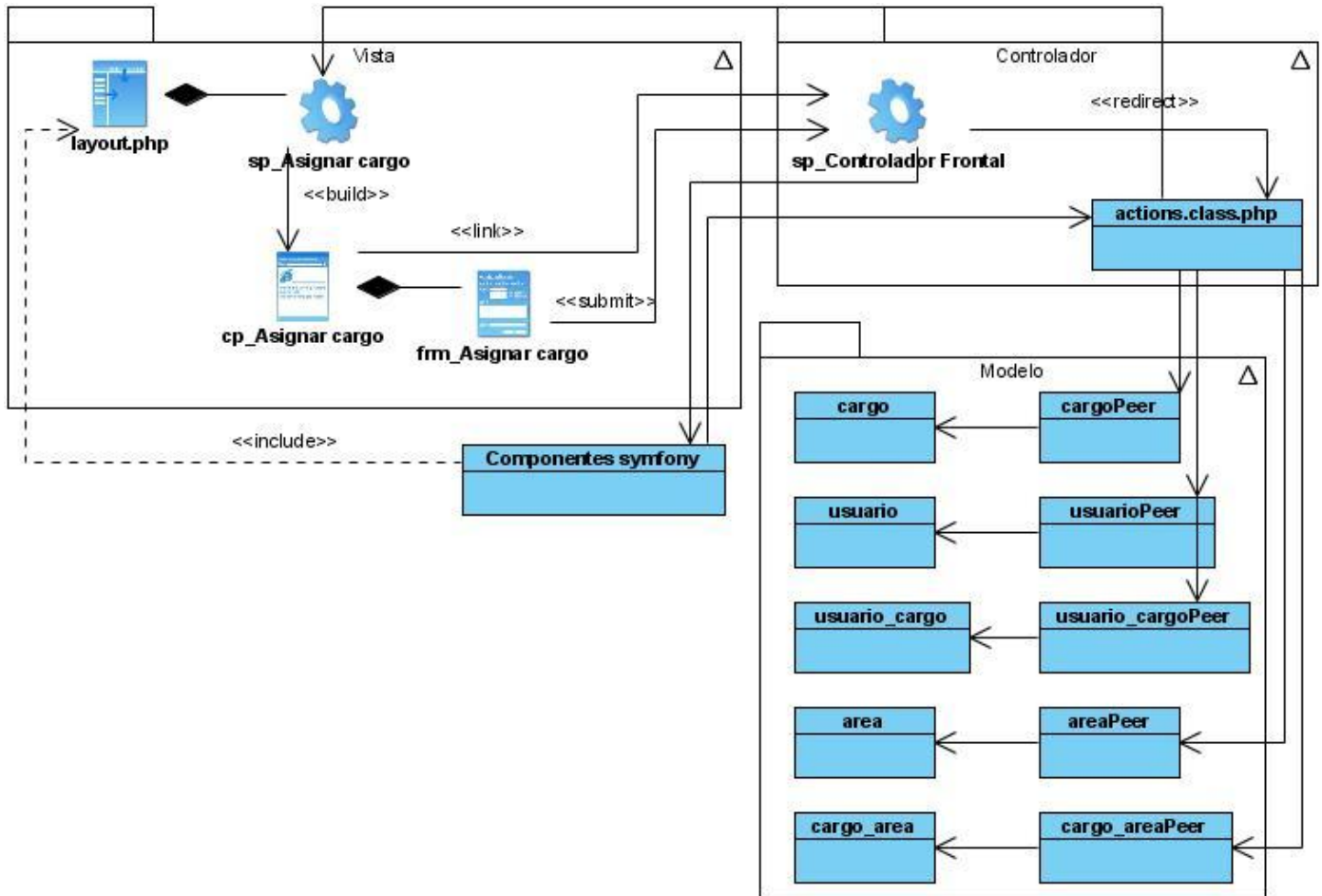


Diagrama de clases del diseño del caso de uso "Asignar cargos"

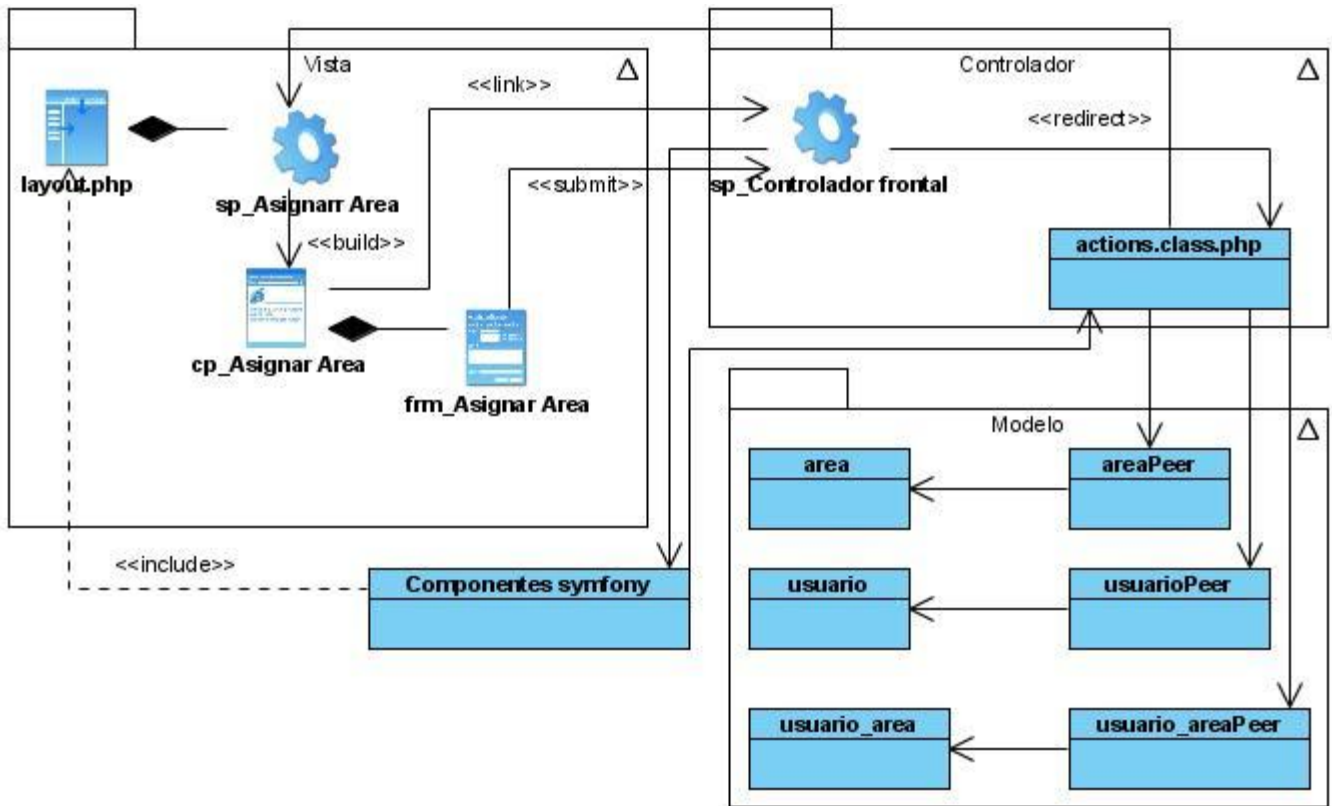


Diagrama de clases del diseño del caso de uso "Gestionar asignación de áreas"

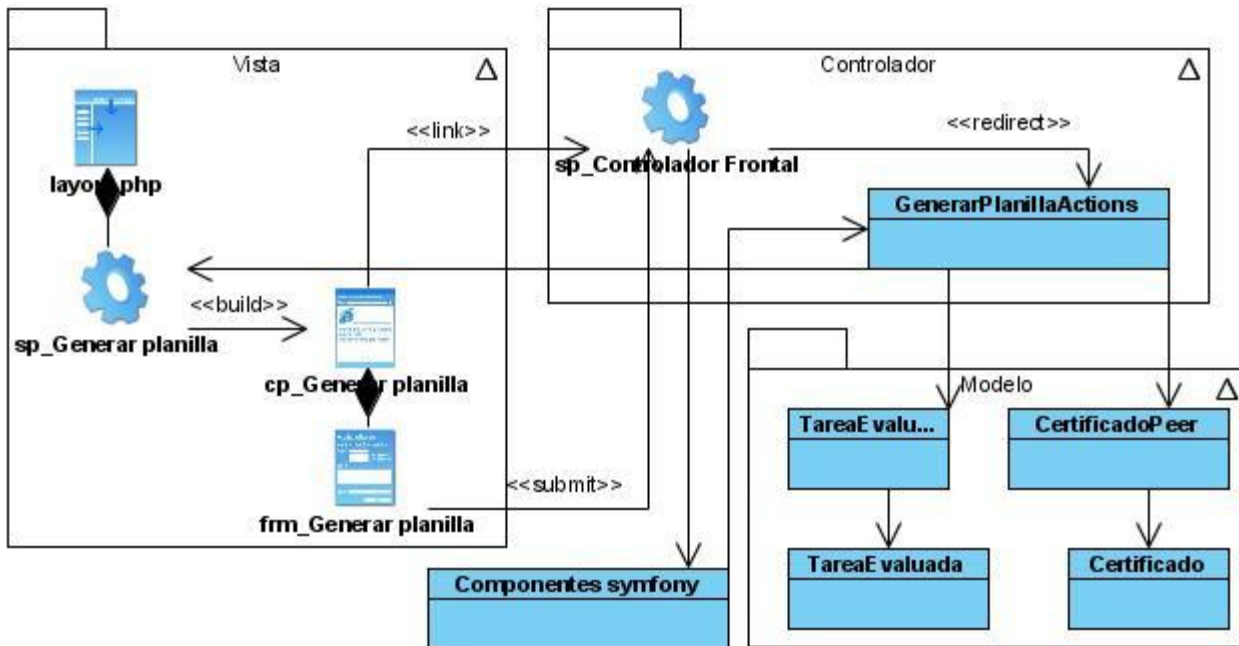


Diagrama de clases del diseño del caso de uso "Generar planilla de evaluación"

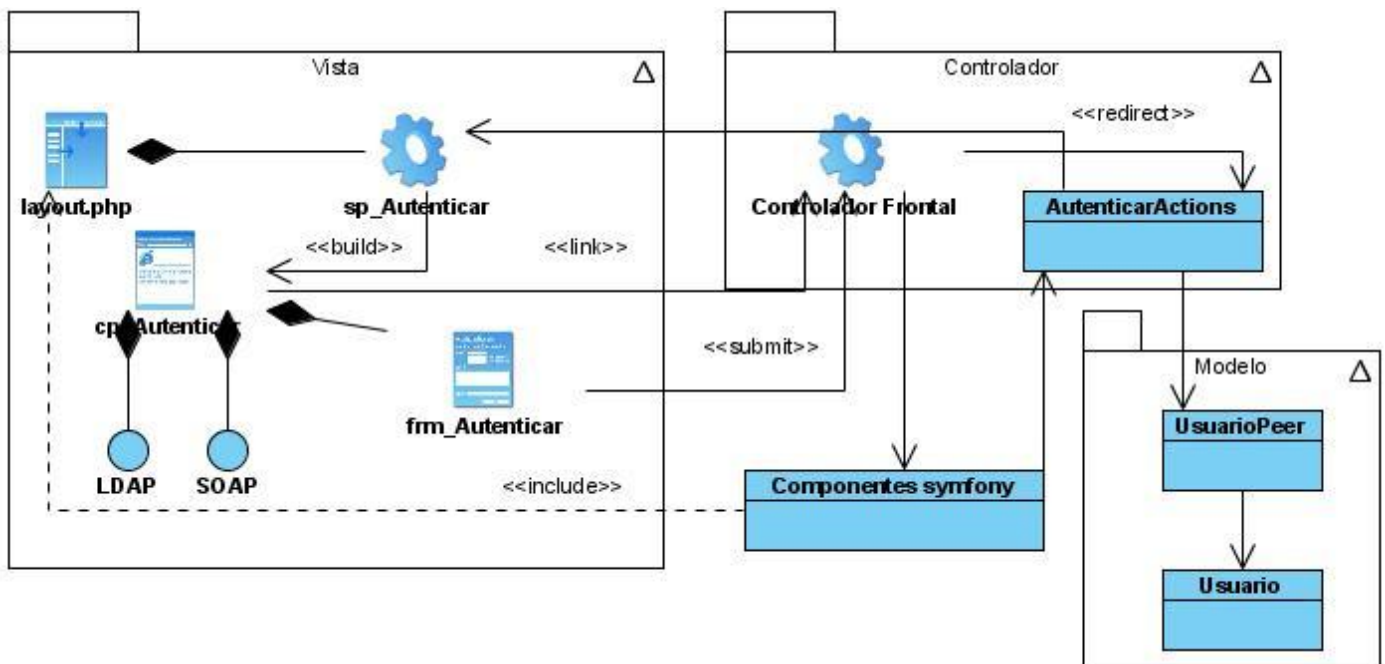


Diagrama de clases del diseño del caso de uso "Autenticar usuario"

Anexo 5: Diagramas de componentes.

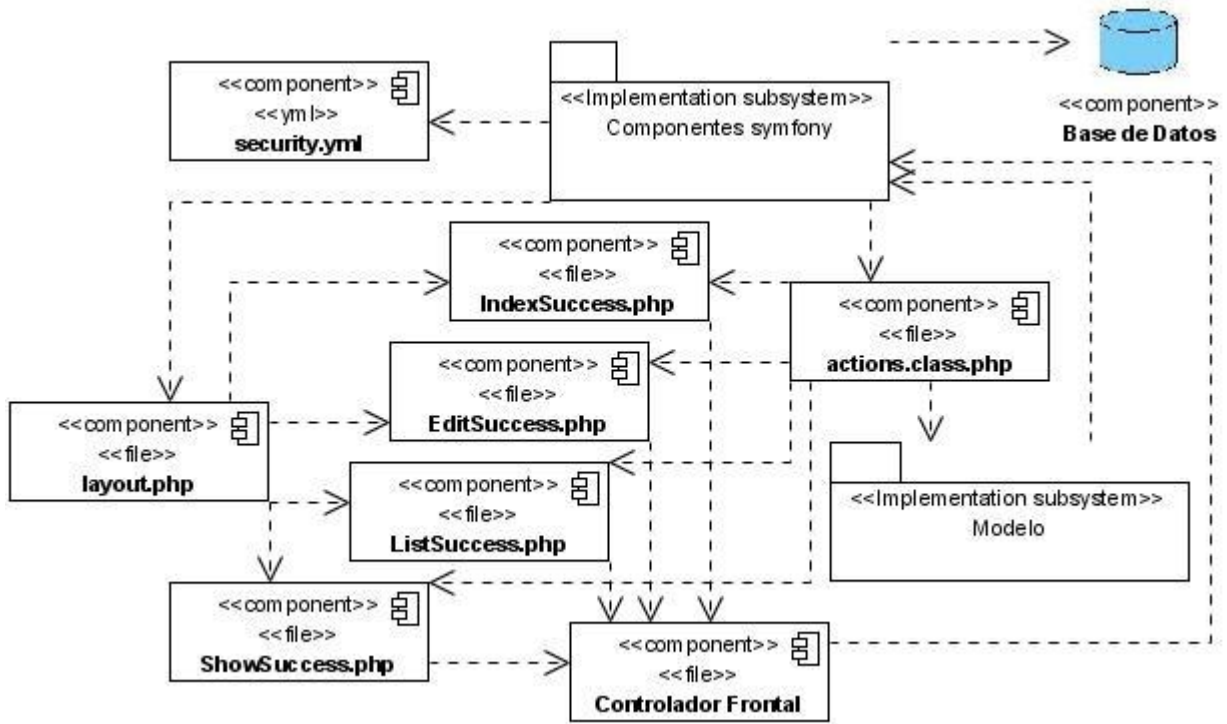


Diagrama de componentes del módulo “gest\_certificado”

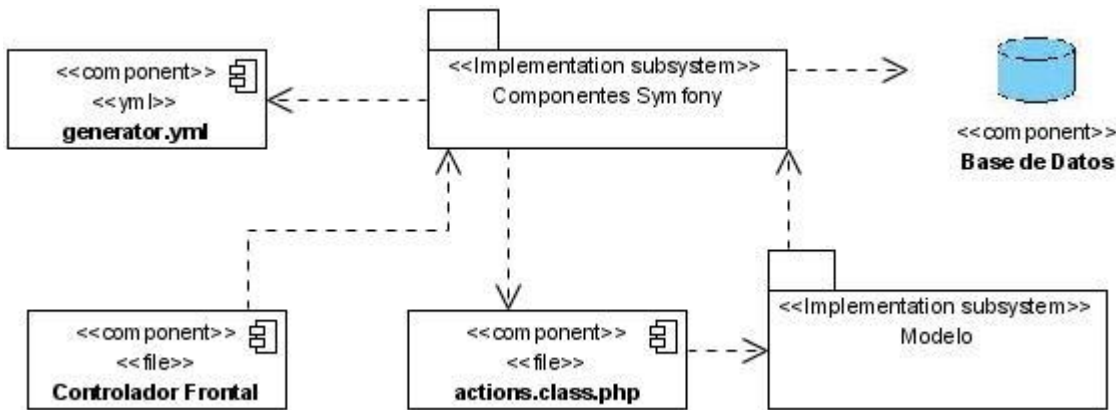


Diagrama de componentes del módulo “funcionalidades”

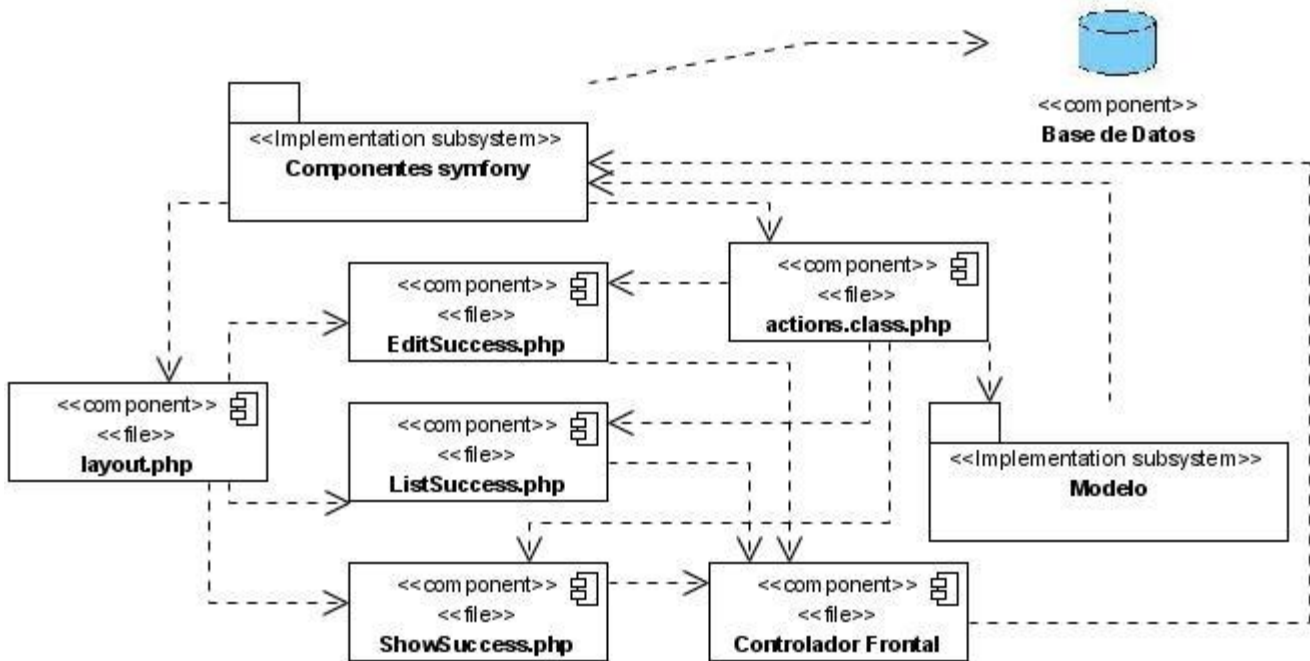


Diagrama de componentes del módulo “gest\_cargo\_area”

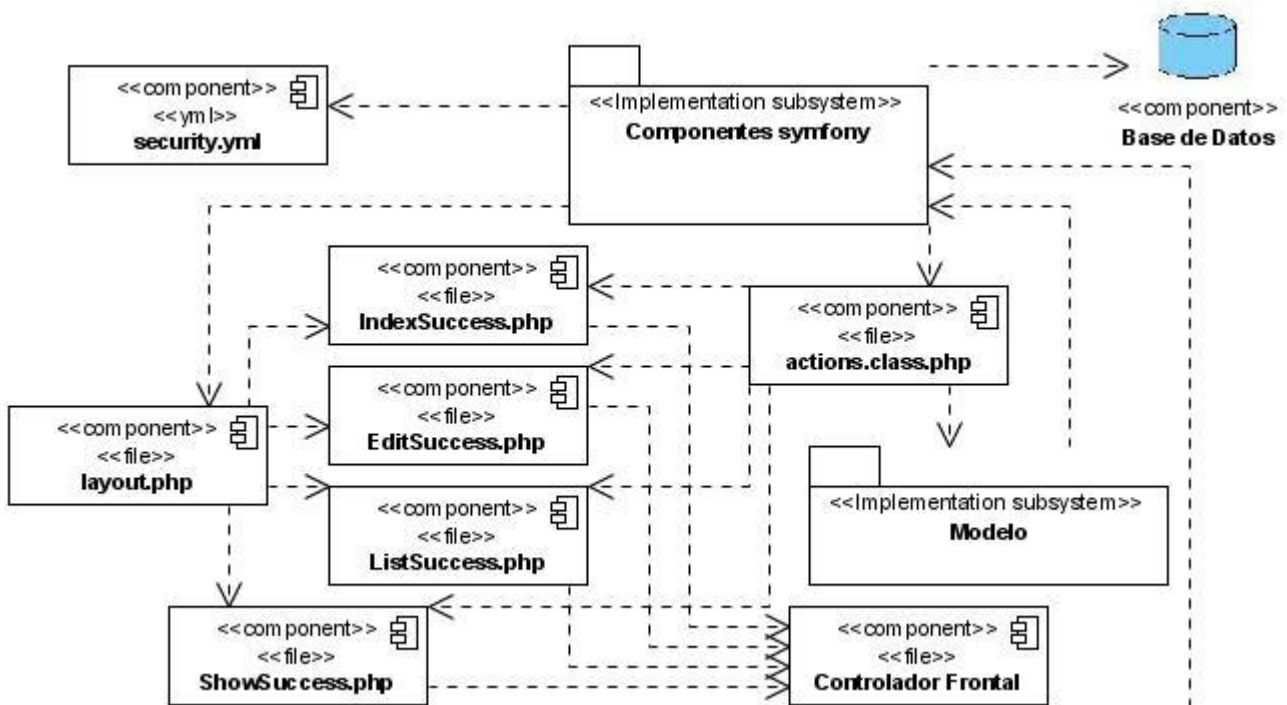


Diagrama de componentes del módulo “gest\_tipo\_evaluacion”

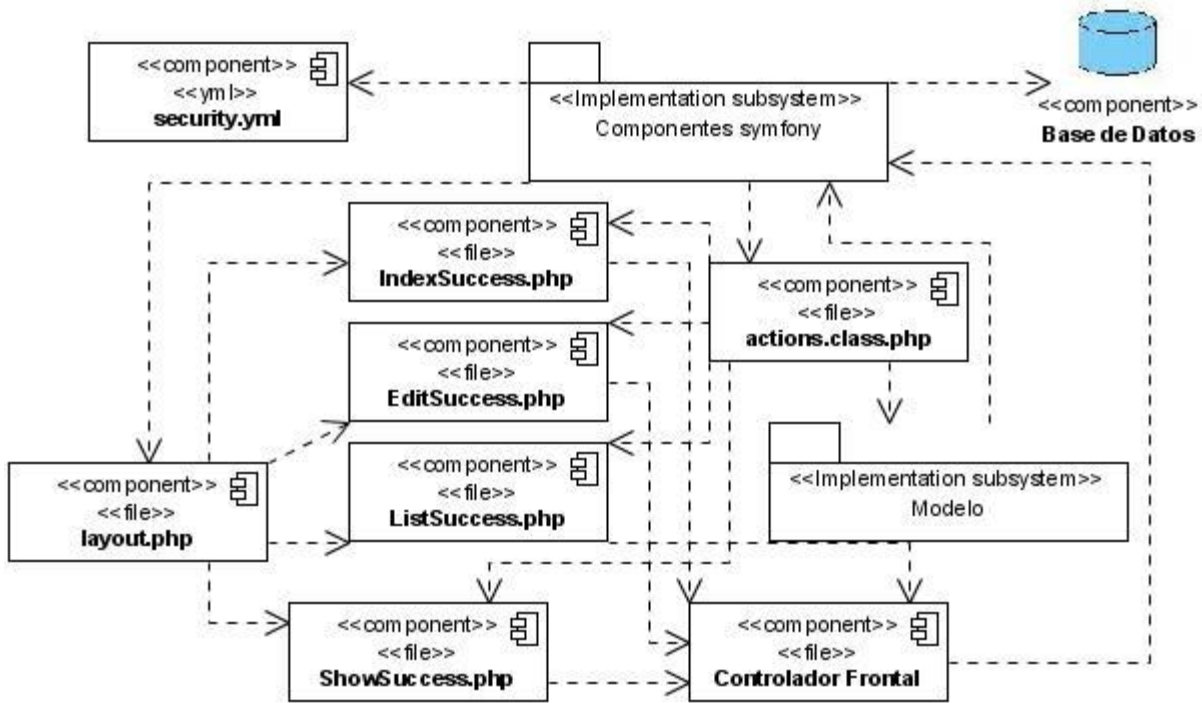


Diagrama de componentes del módulo "gest\_usuario\_local"

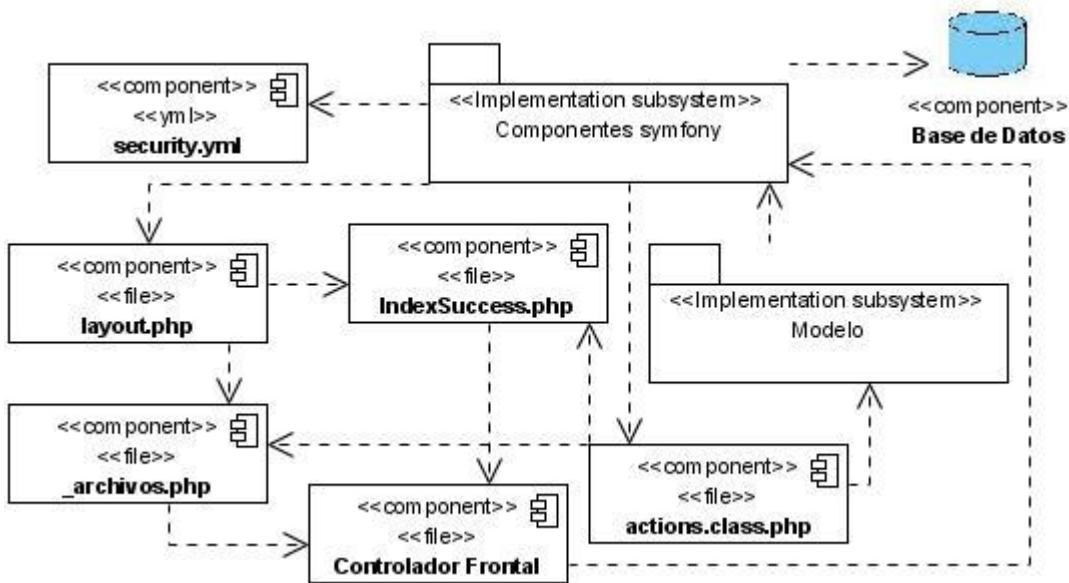


Diagrama de componentes del módulo "gest\_informacion"



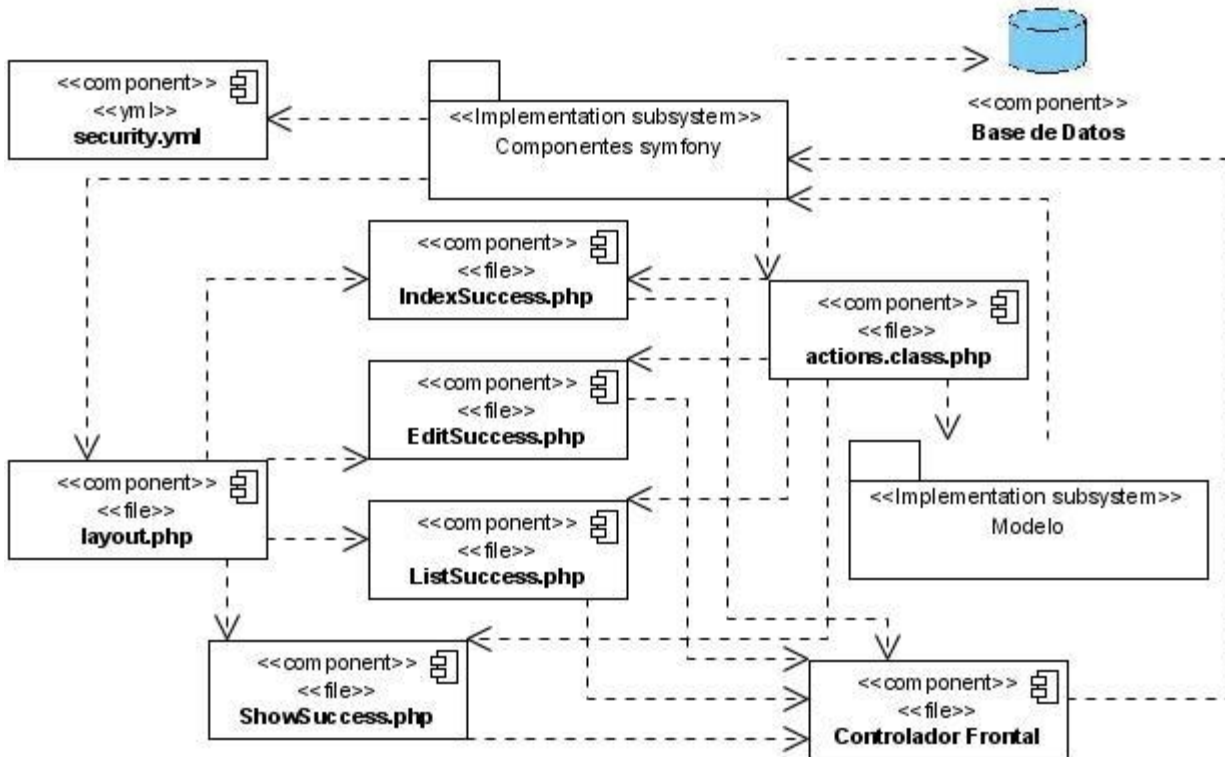


Diagrama de componentes del módulo “gest\_parametro”

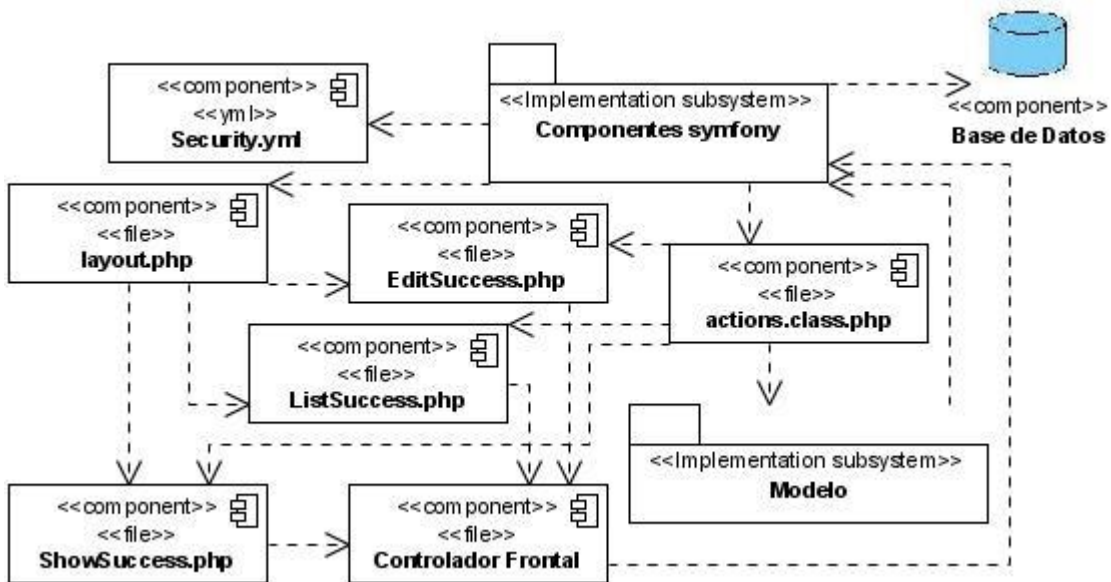


Diagrama de componentes del módulo “gest\_tarea\_area”



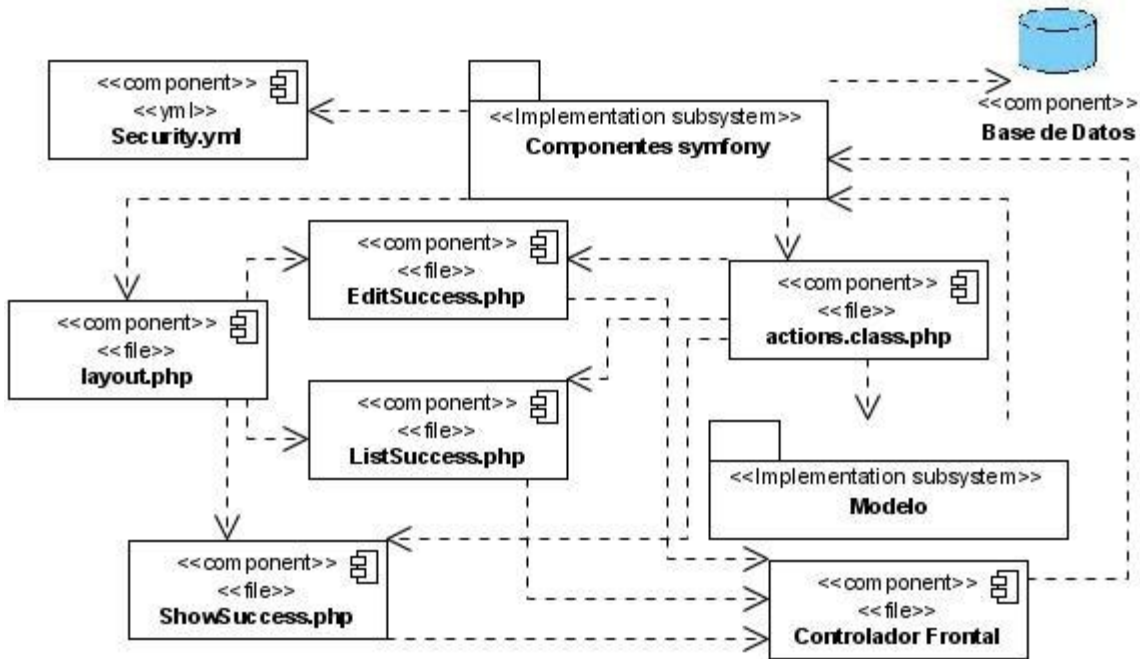


Diagrama de componentes del módulo “gest\_tarea\_cargo”

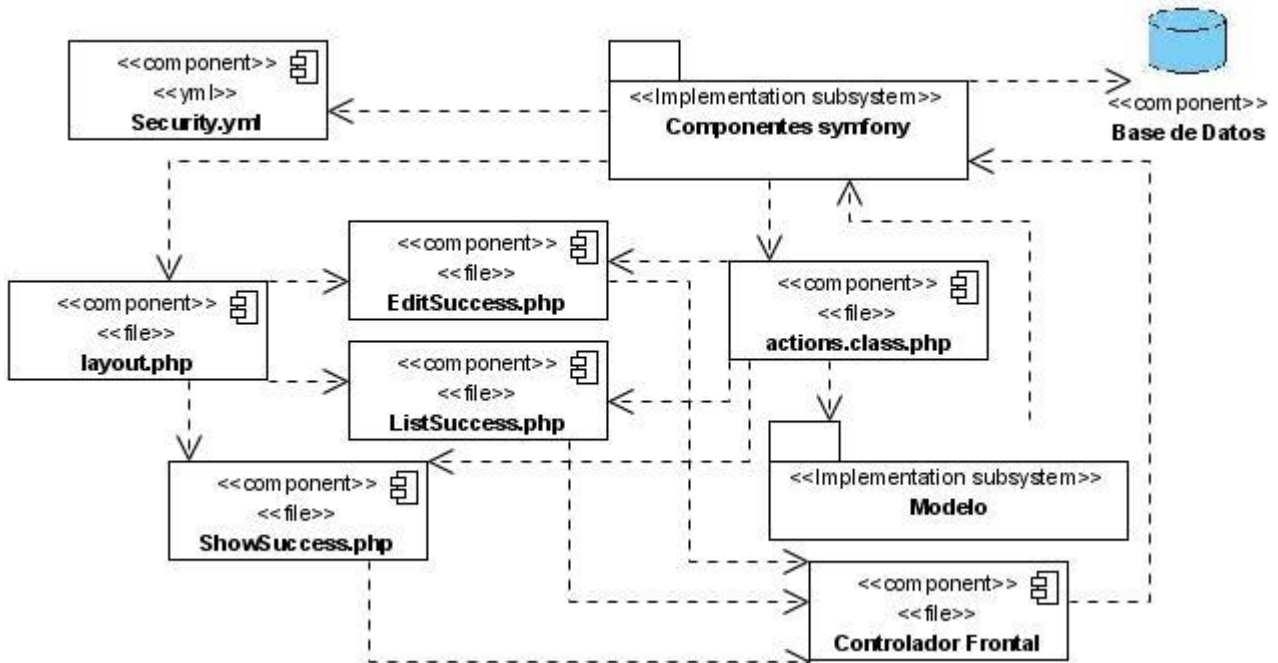


Diagrama de componentes del módulo “gest\_tarea\_cargo\_area”

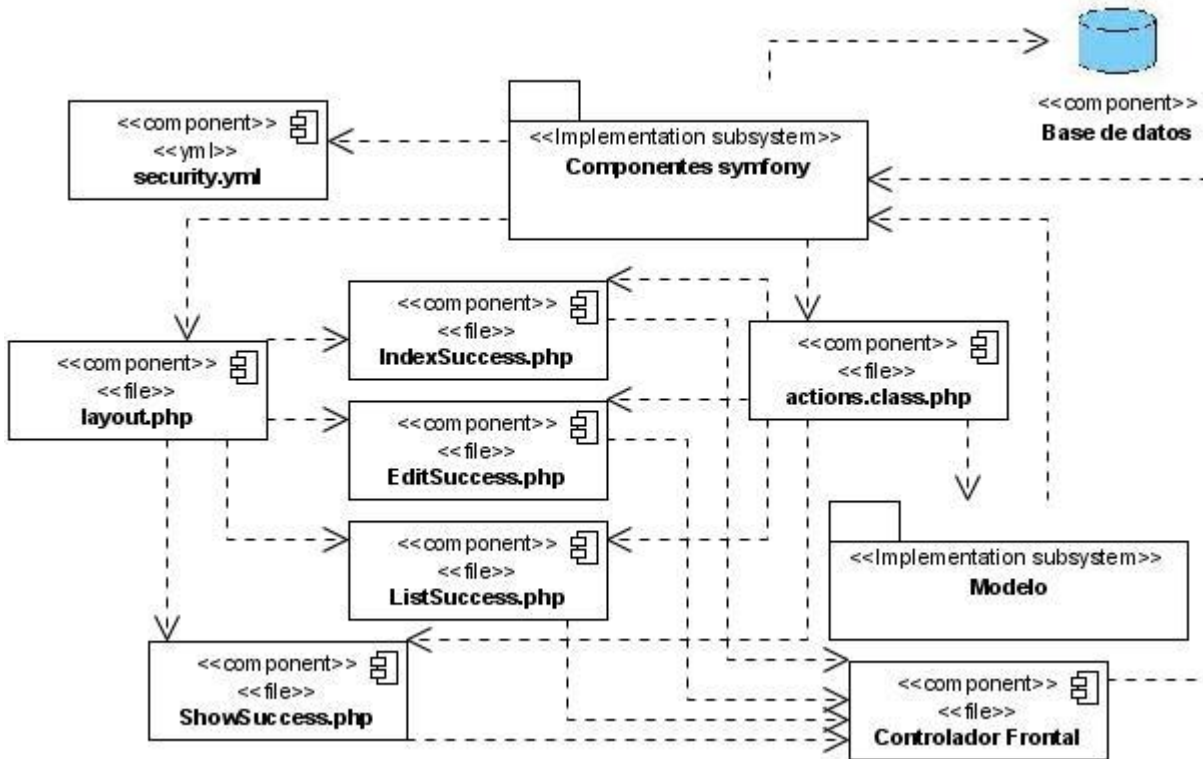


Diagrama de componentes del módulo “gest\_tarea\_especial”

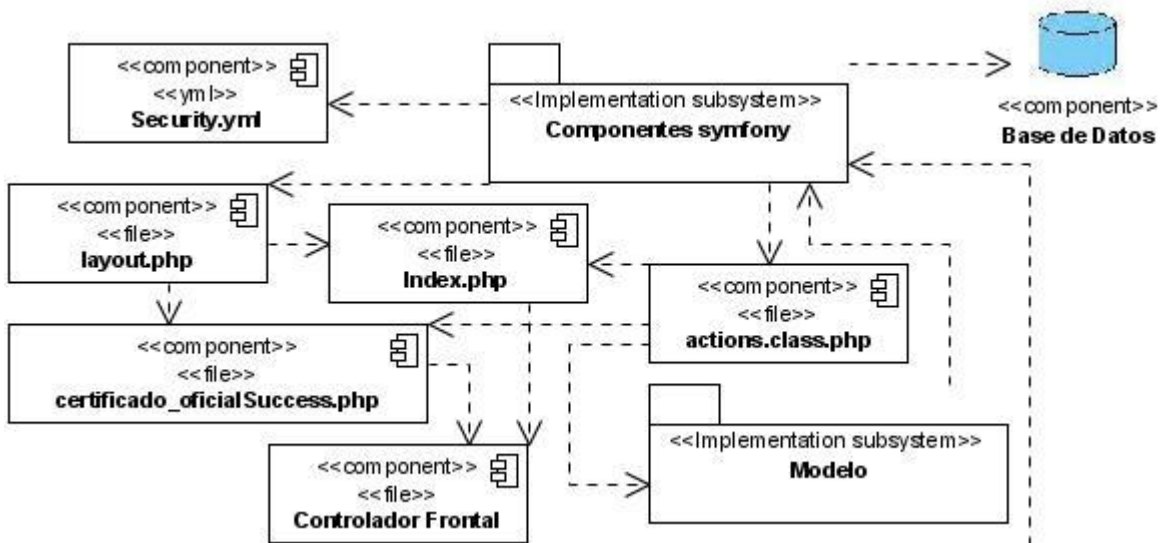


Diagrama de componentes del módulo “reportes”

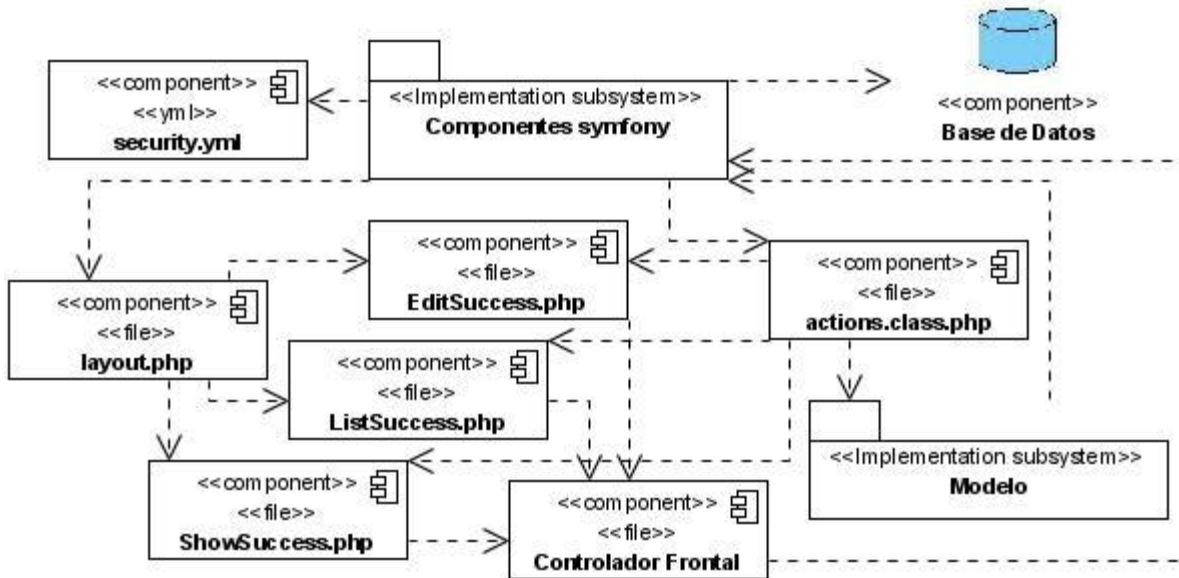


Diagrama de componentes del módulo “gest\_evaluador\_evaluado”

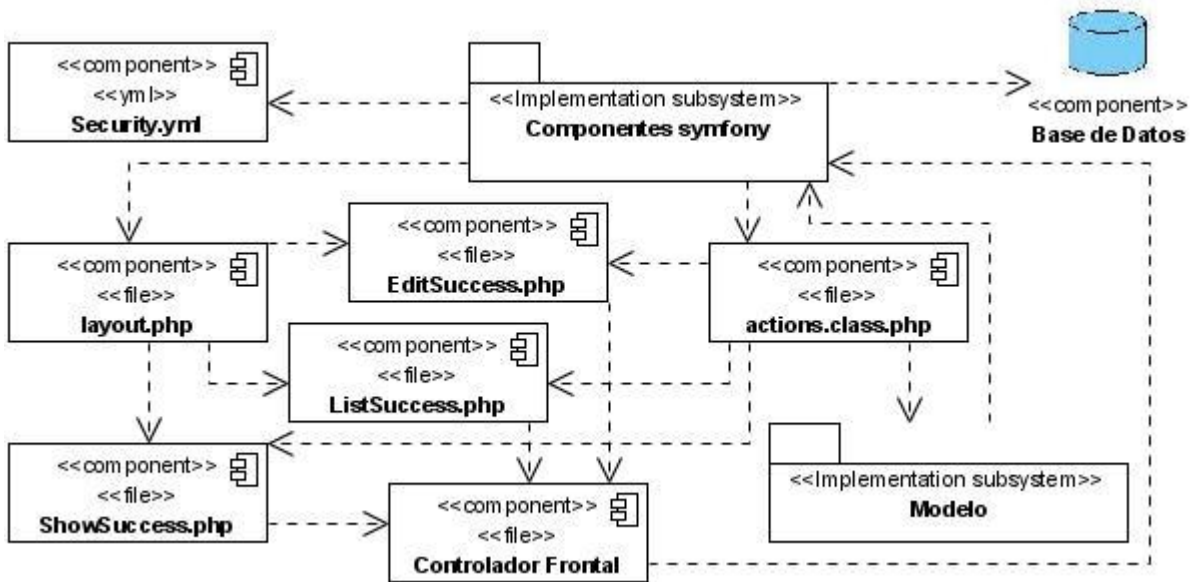


Diagrama de componentes del módulo “gest\_evaluador\_evaluado\_cargo\_area”

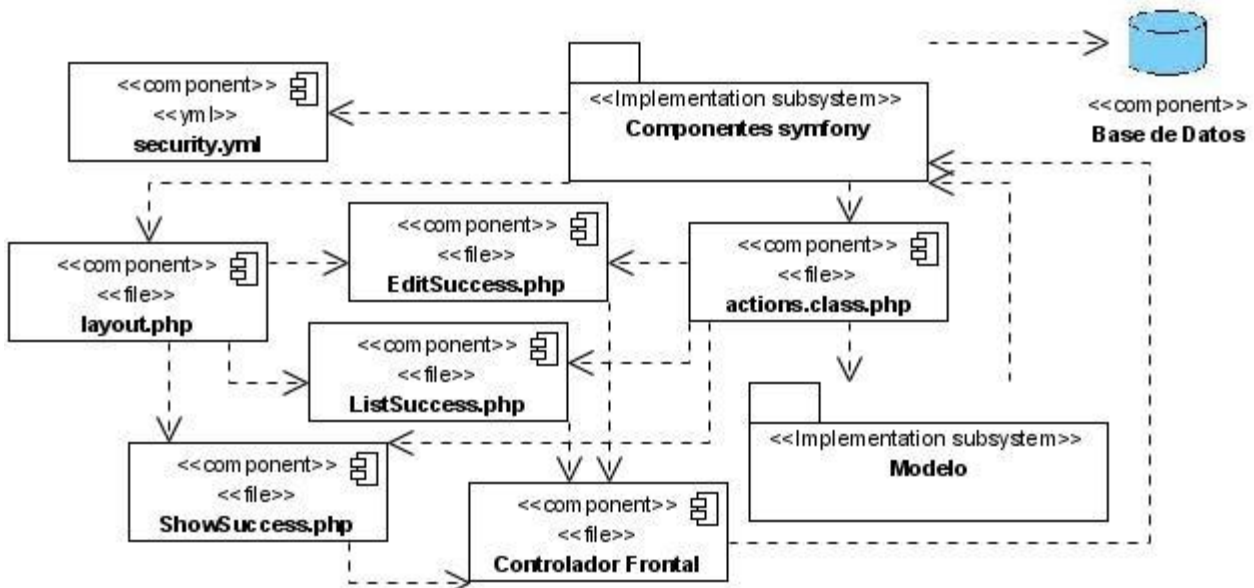


Diagrama de componentes del módulo “gest\_cargo\_funcionalidad”

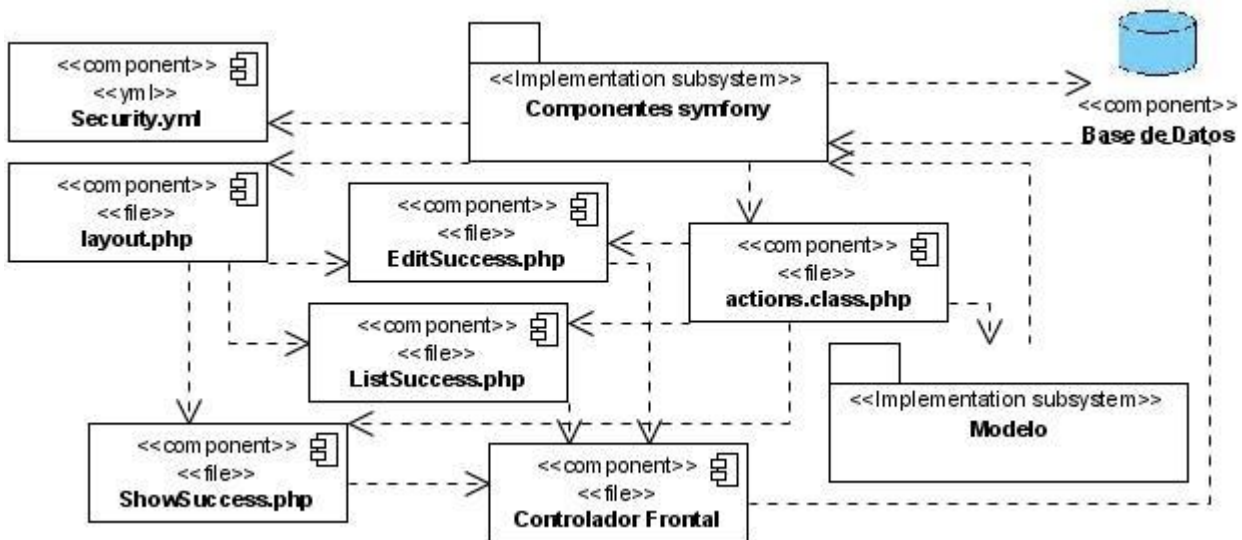


Diagrama de componentes del módulo “gest\_usuario\_cargo”

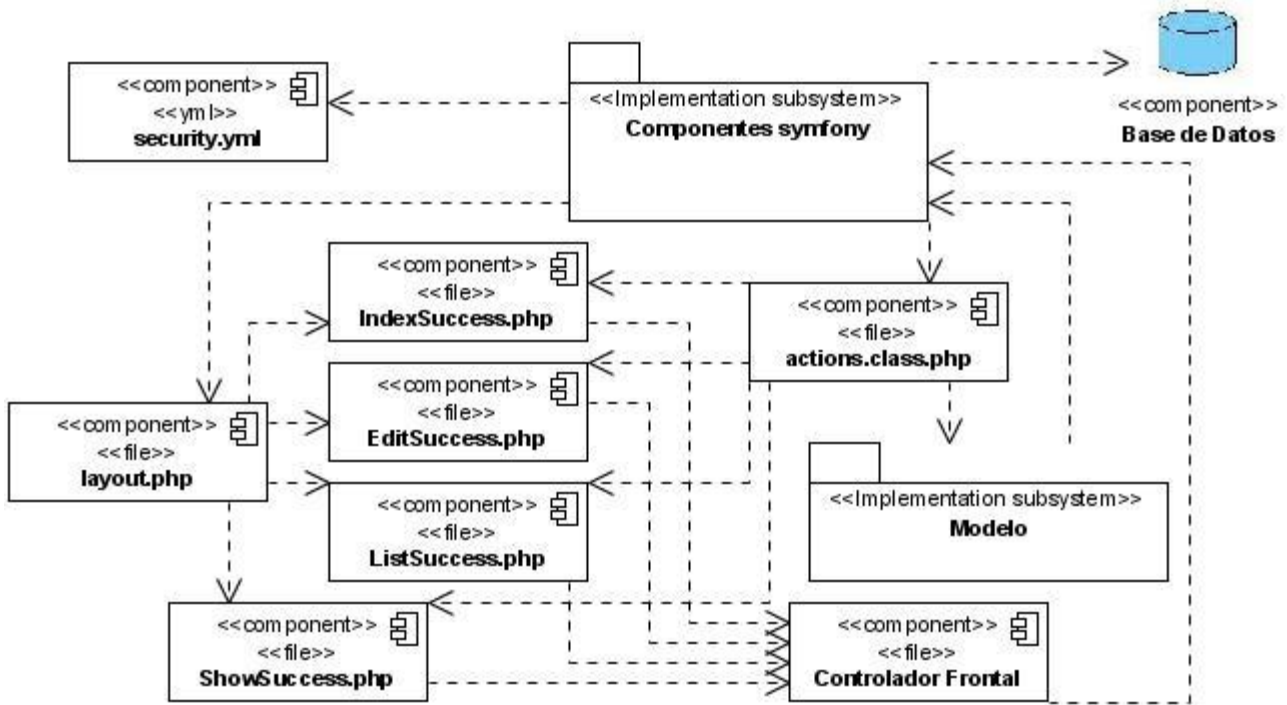


Diagrama de componentes del módulo “gest\_tipo\_de\_evaluacion”

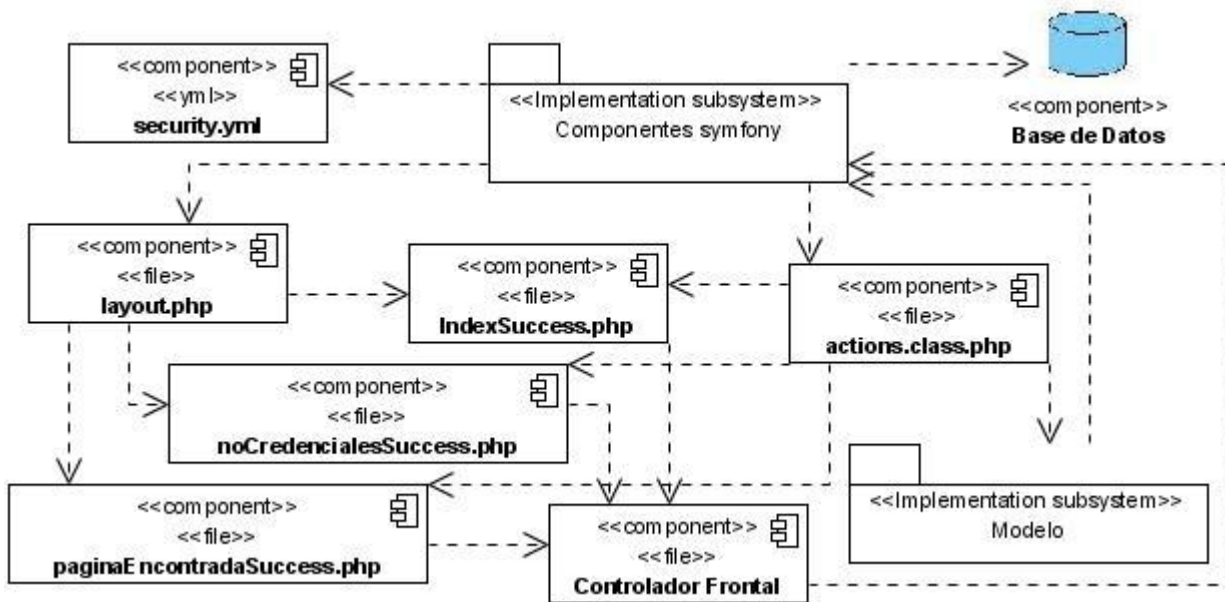


Diagrama de componentes del módulo “error”

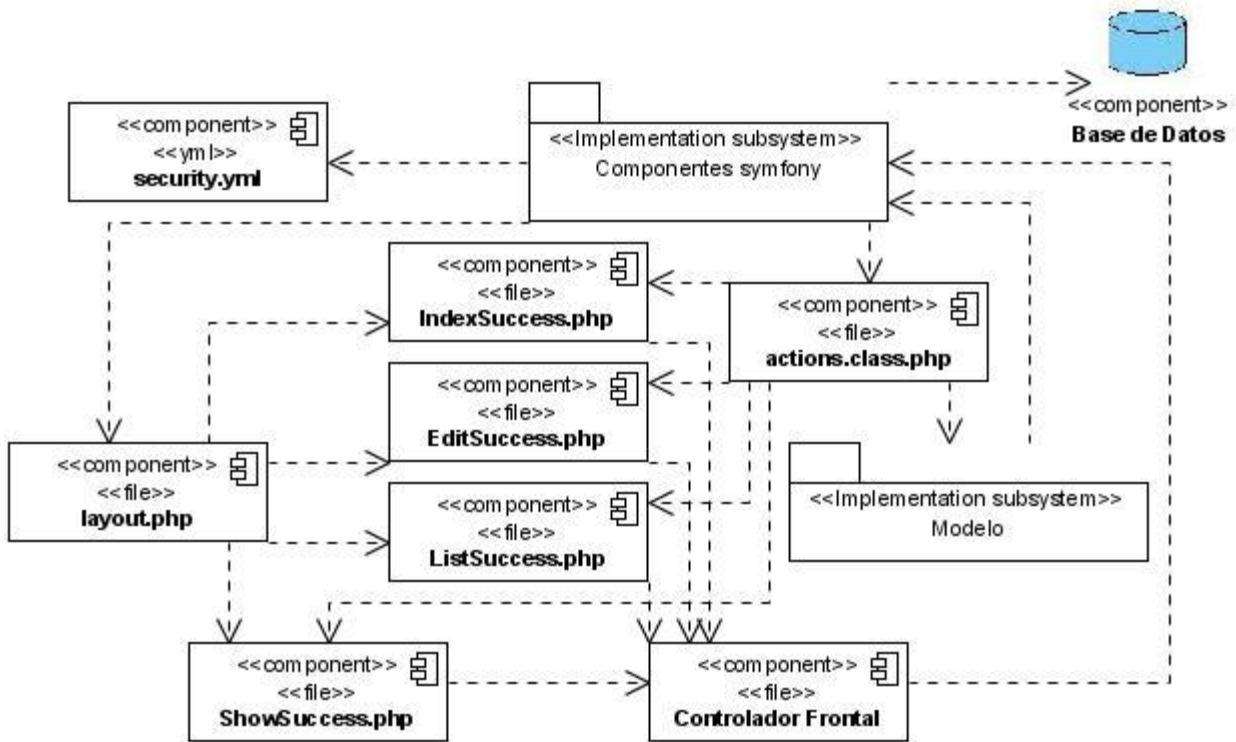


Diagrama de componentes del módulo “gest\_tipo\_de\_tarea”



Anexo 6. Diagramas de secuencia

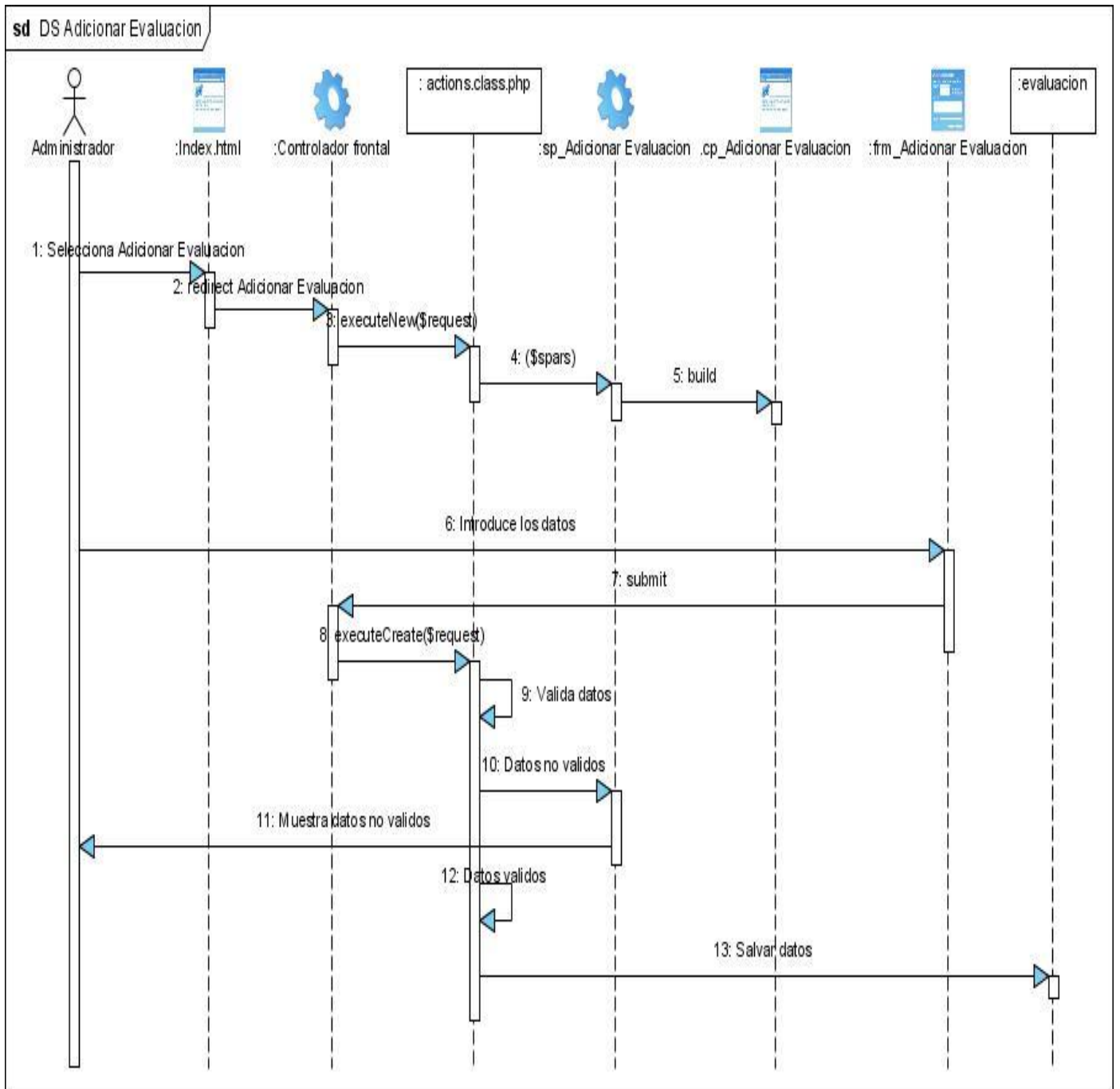


Diagrama de secuencia CU Gestionar evaluación. Escenario “Adicionar evaluación”.

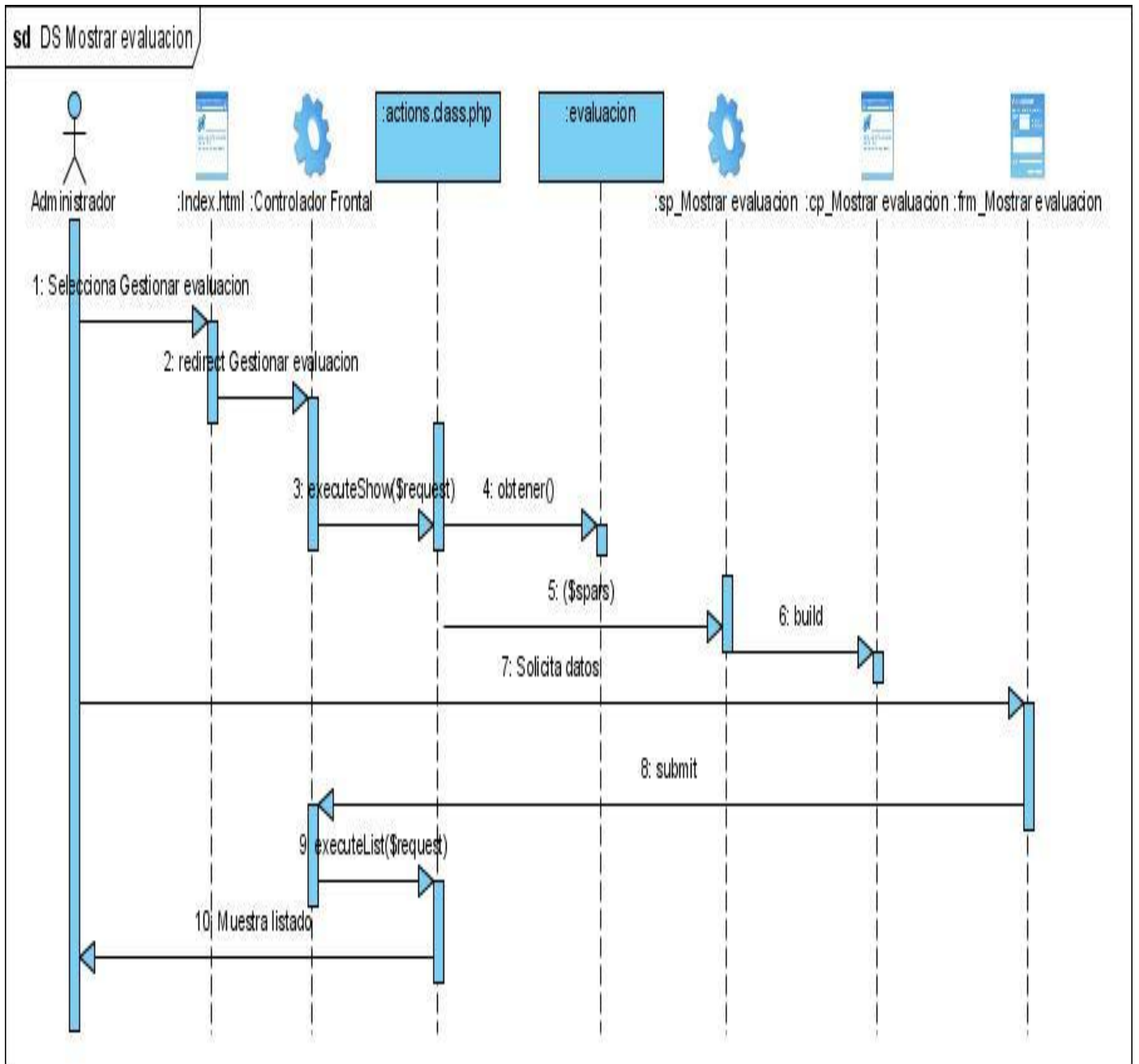


Diagrama de secuencia CU Gestionar evaluación. Escenario “Mostrar evaluación”.



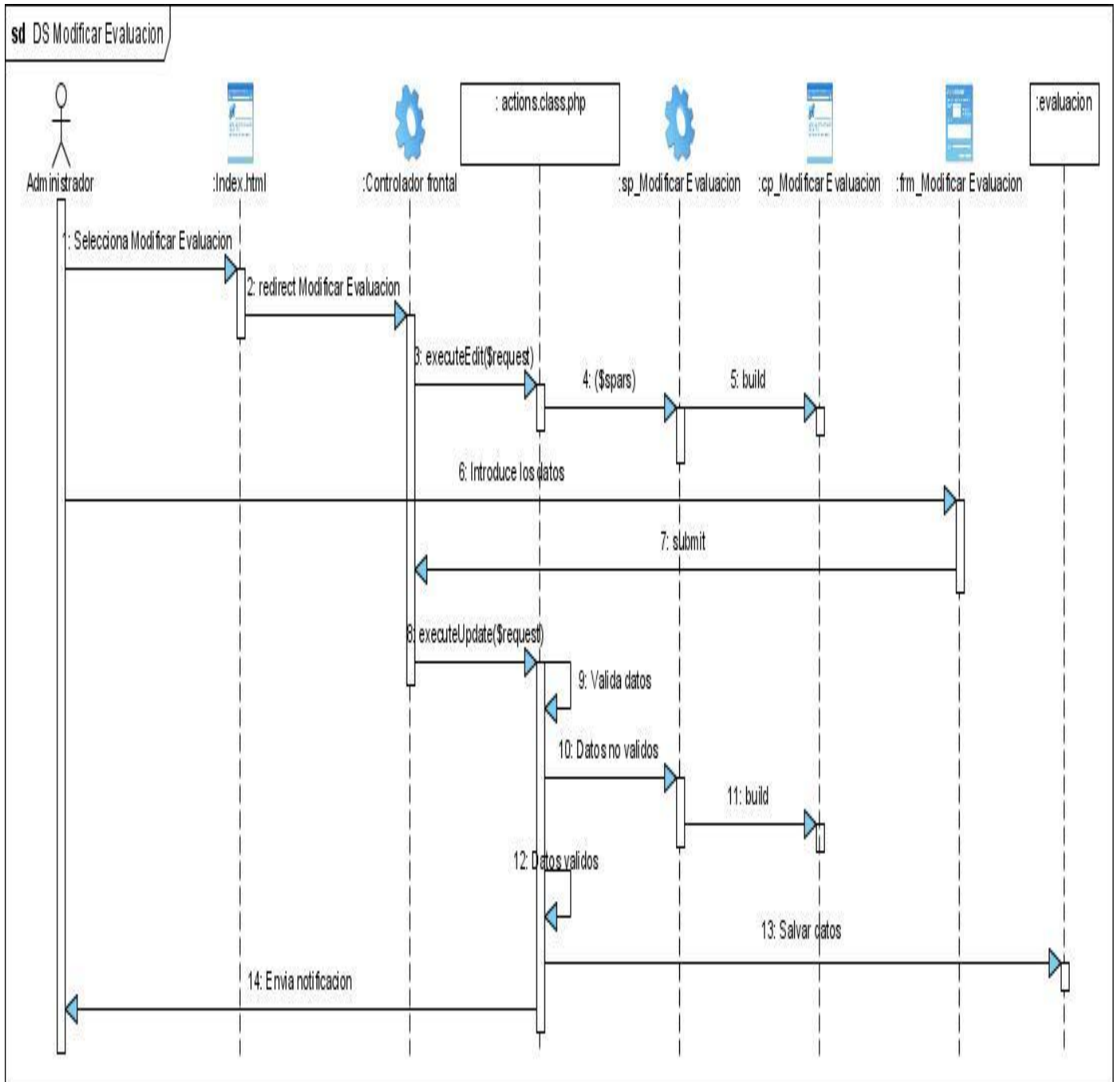


Diagrama de secuencia CU Gestionar evaluación. Escenario “Modificar evaluación”.

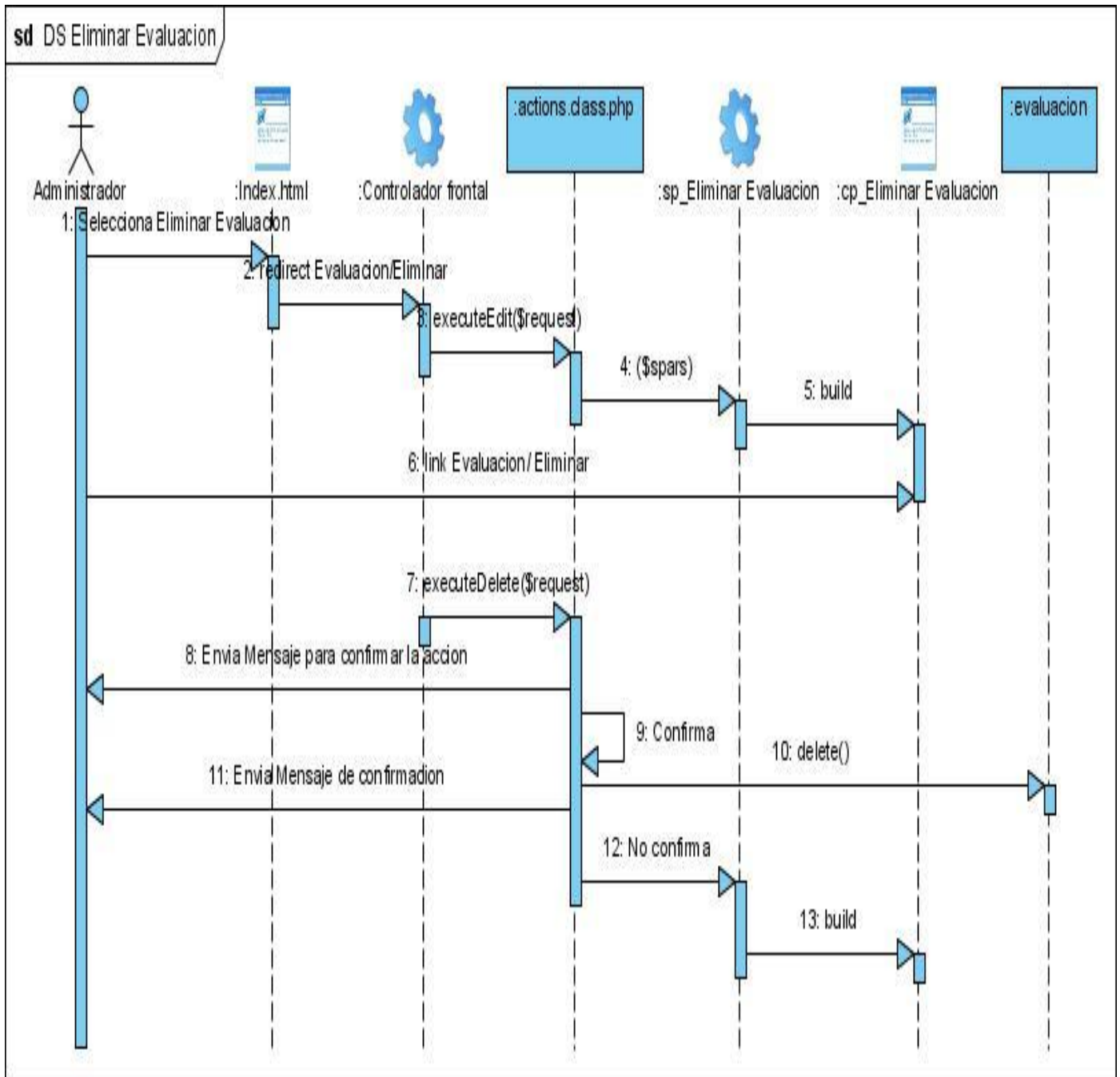


Diagrama de secuencia CU Gestionar evaluación. Escenario “Eliminar evaluación”.

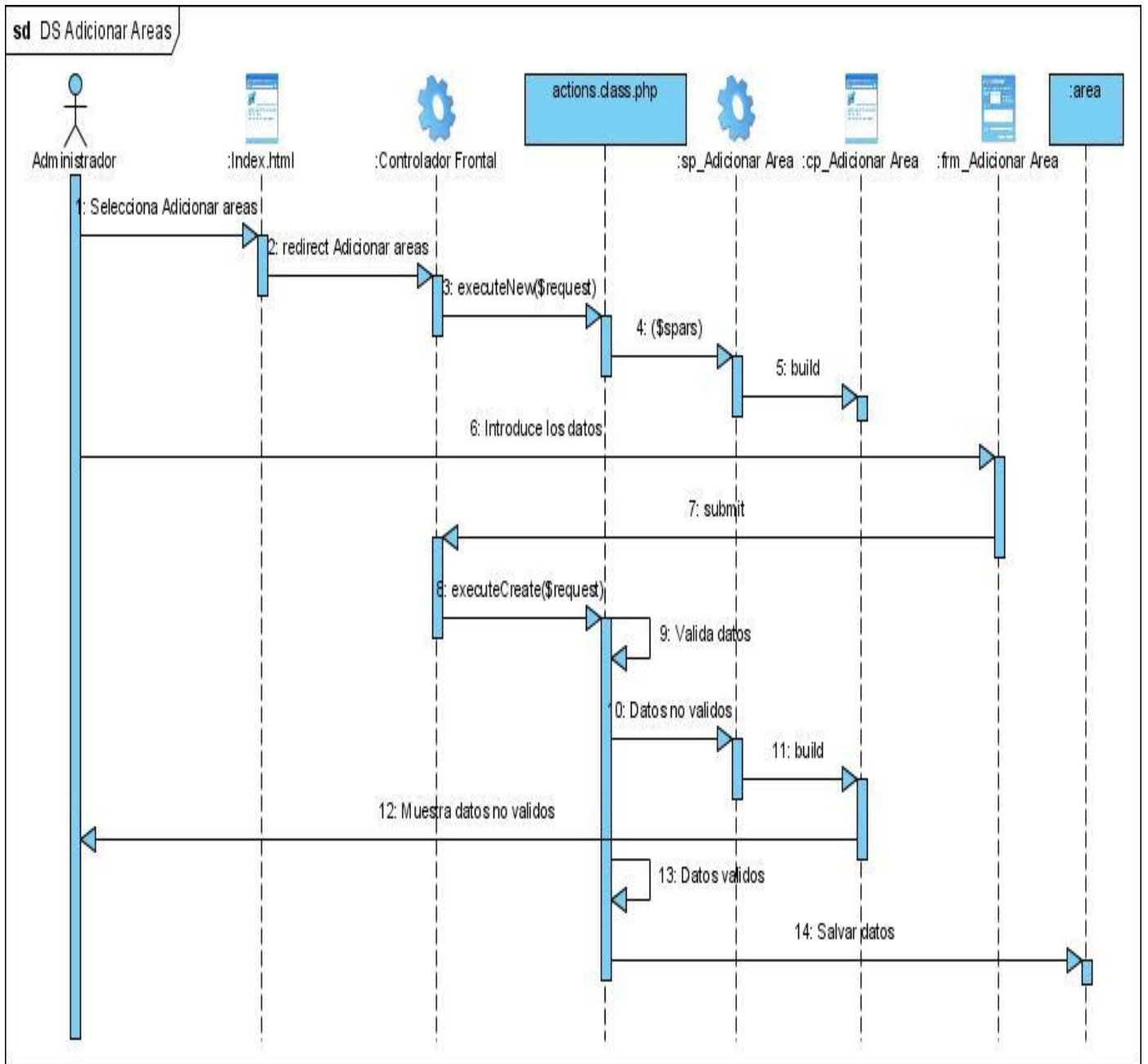


Diagrama de secuencia CU Gestionar área. Escenario “Adicionar área”.

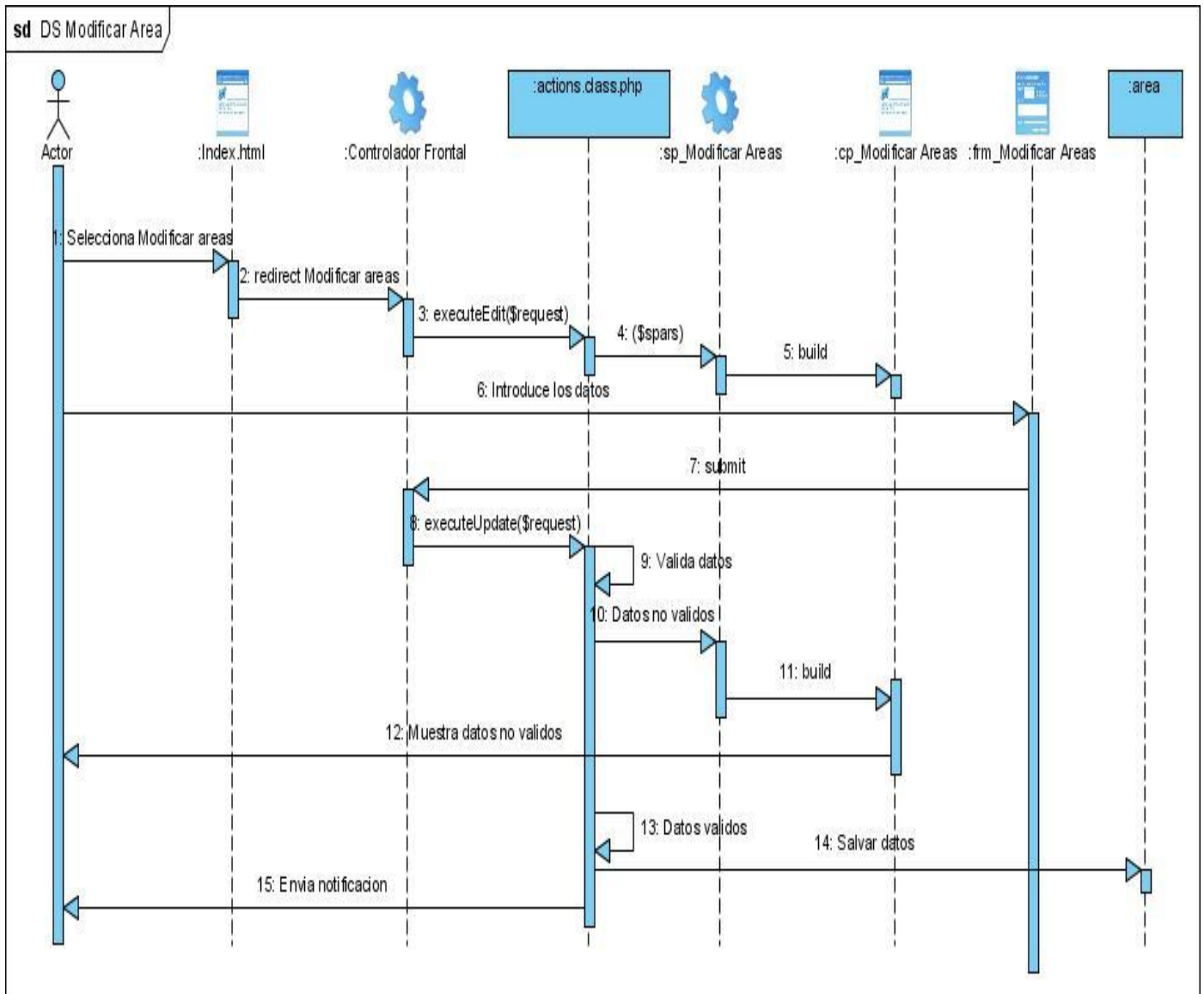


Diagrama de secuencia CU Gestionar área. Escenario “Modificar área”.

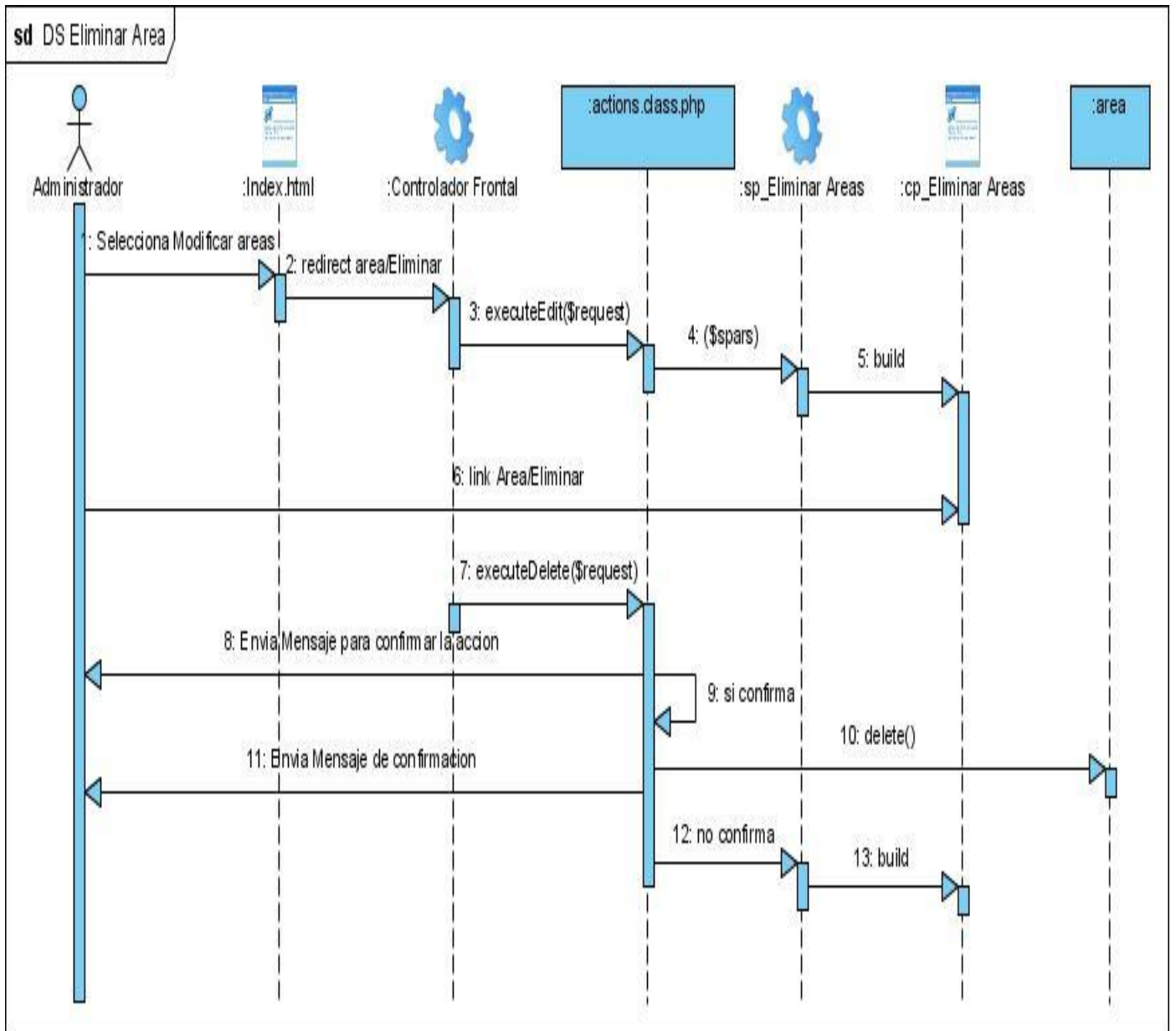


Diagrama de secuencia CU Gestionar área. Escenario “Eliminar área”.

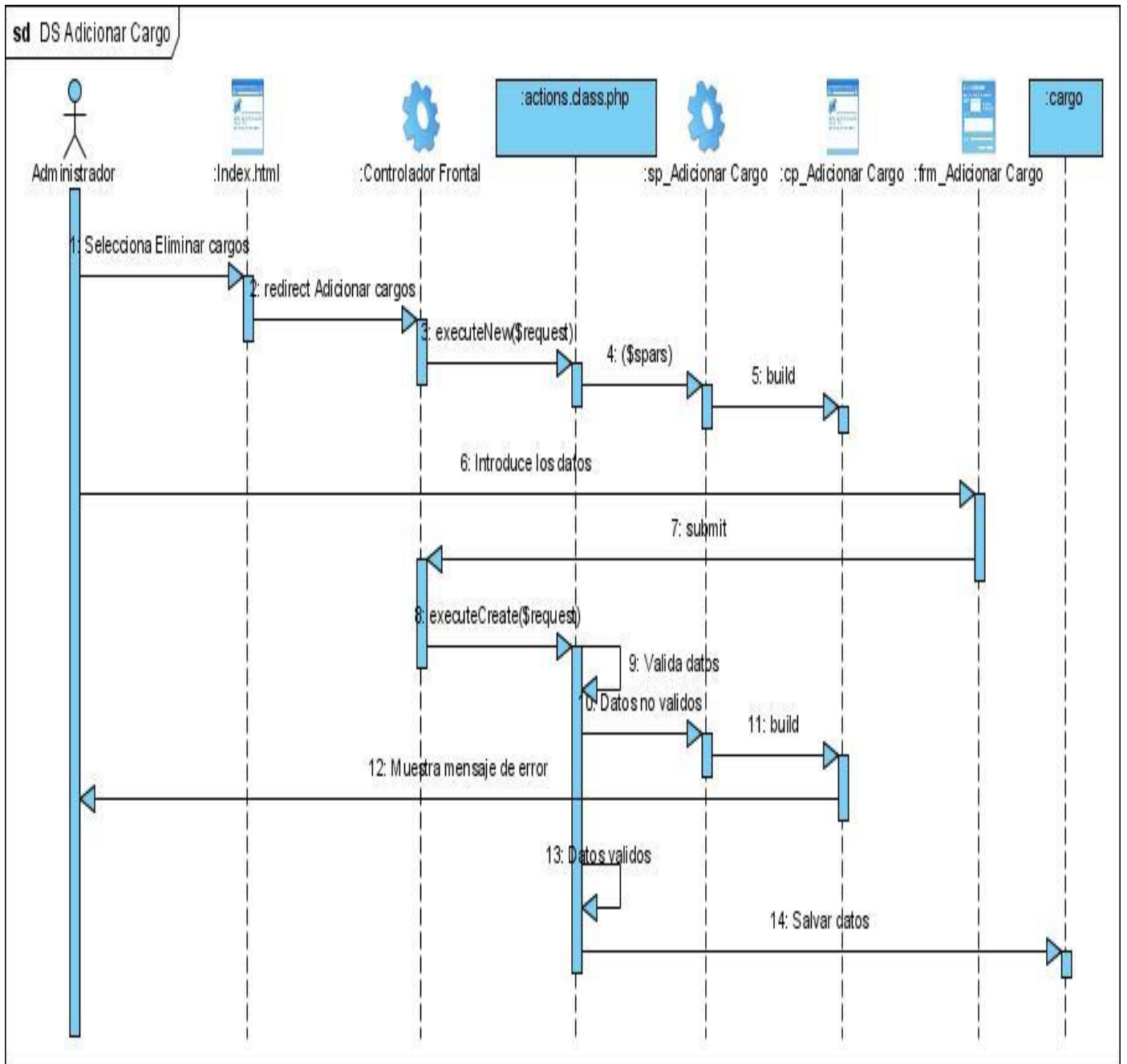


Diagrama de secuencia CU Gestionar cargo. Escenario “Adicionar cargo”.

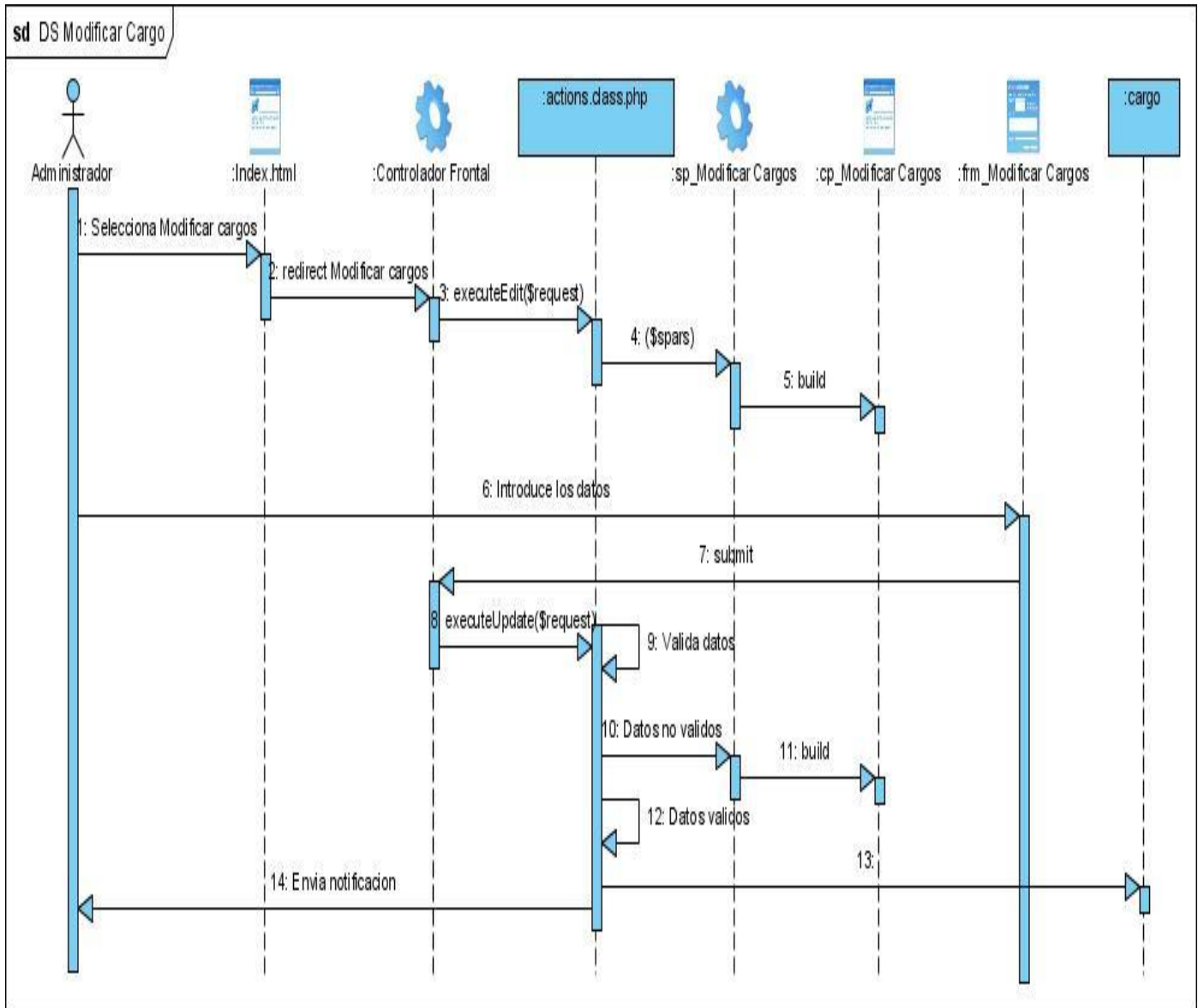


Diagrama de secuencia CU Gestionar cargo. Escenario “Modificar cargo”.

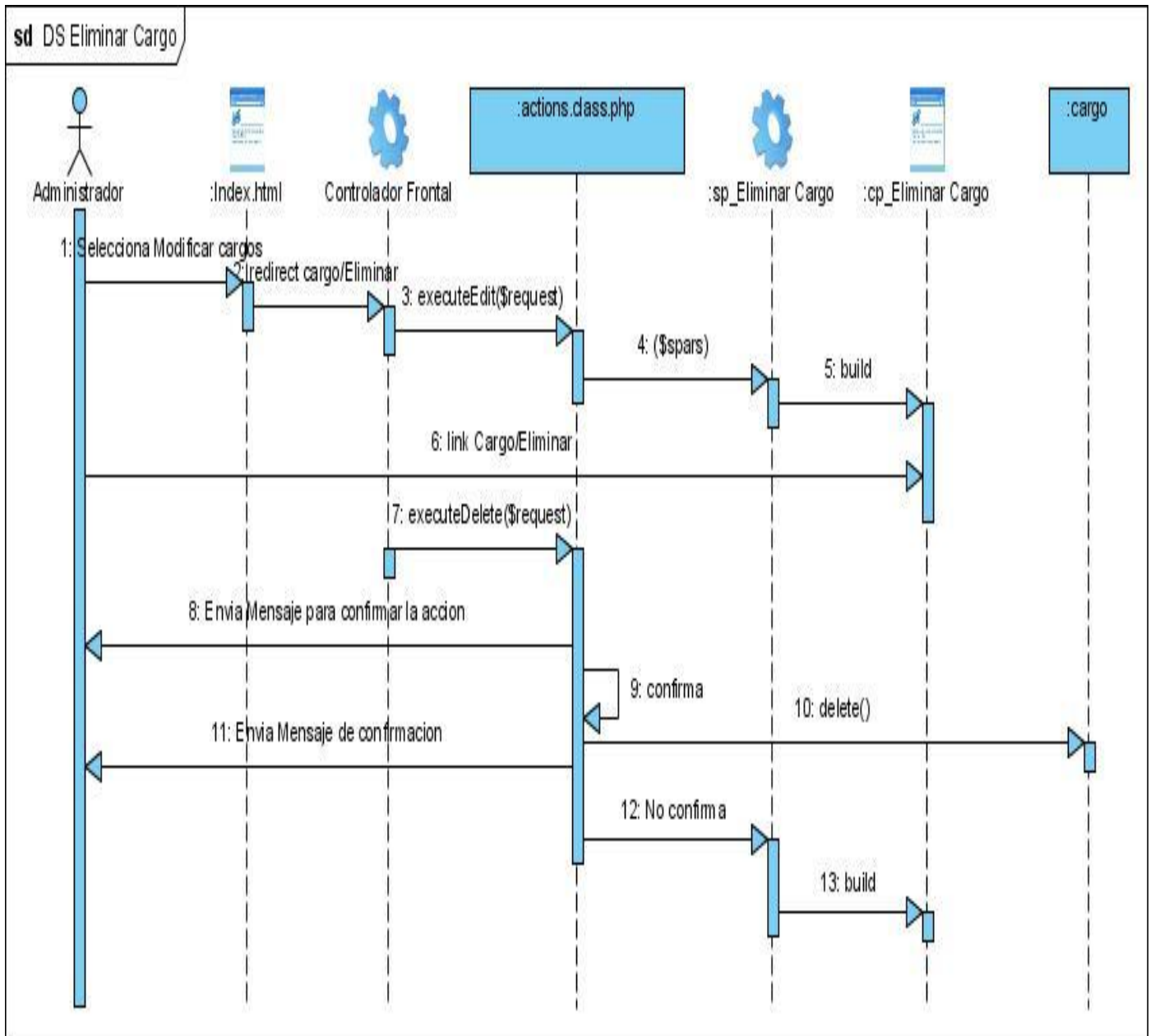


Diagrama de secuencia CU Gestionar cargo. Escenario “Eliminar cargo”.



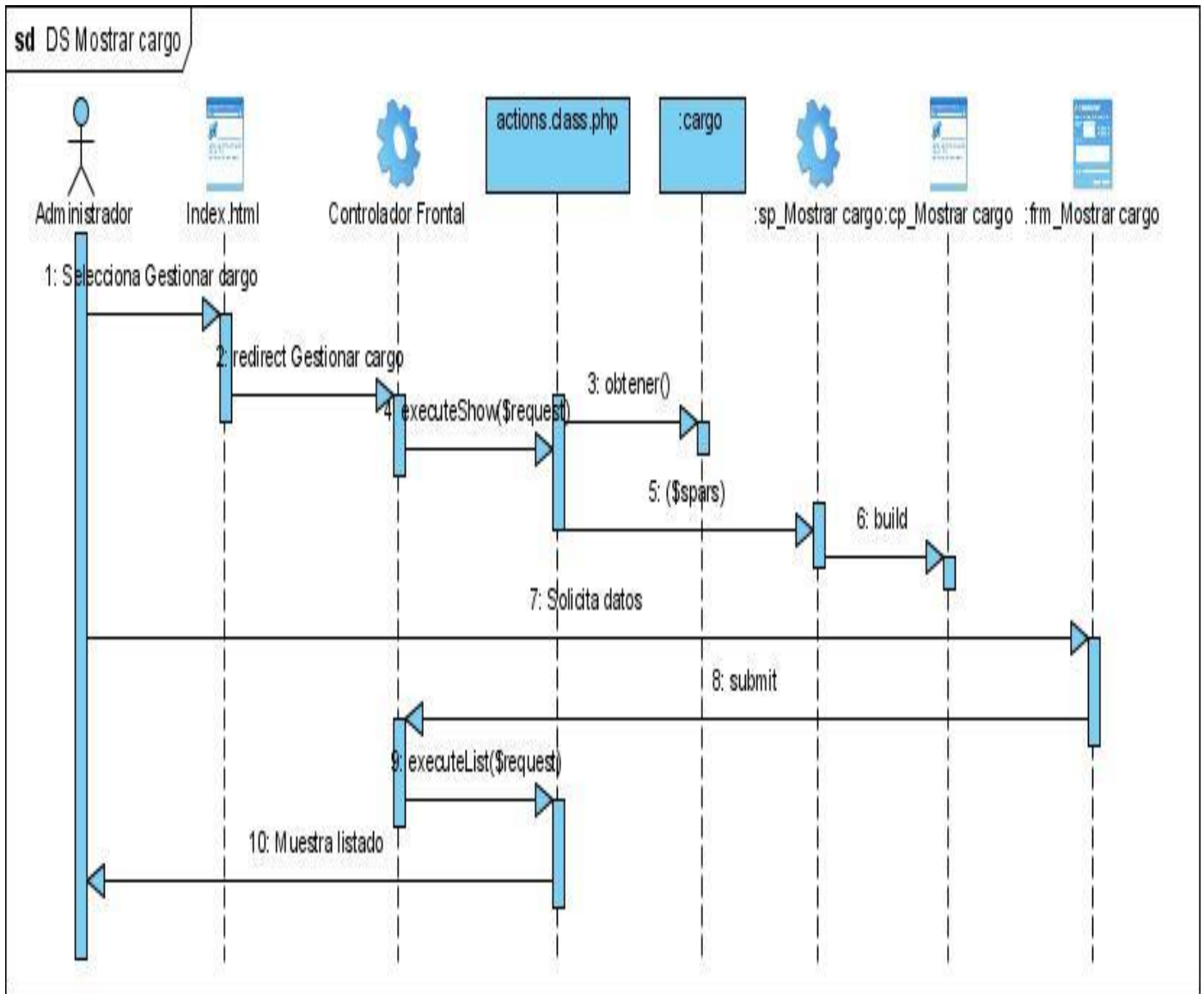


Diagrama de secuencia CU Gestionar cargo. Escenario “Mostrar cargo”.

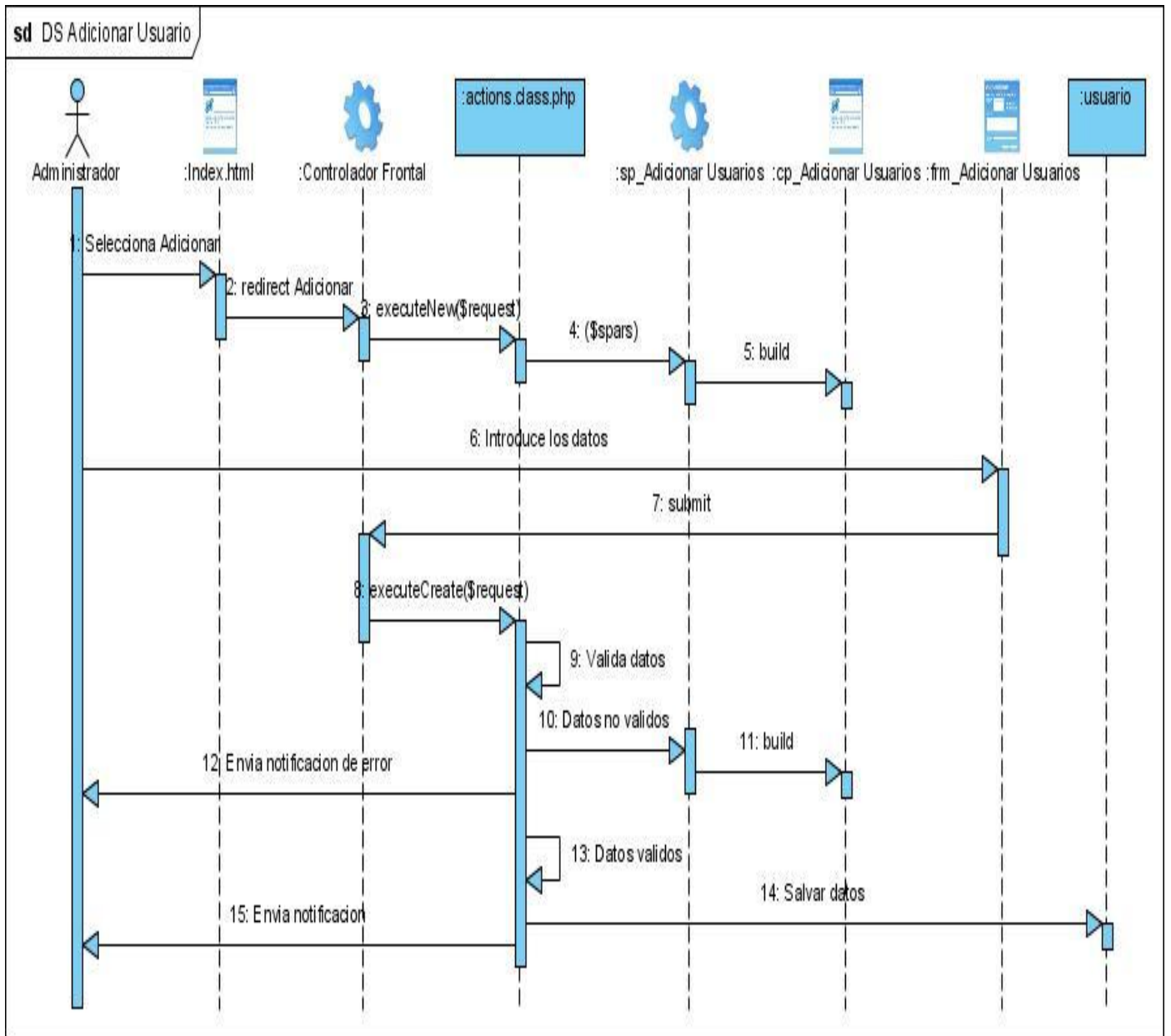


Diagrama de secuencia CU Gestionar usuario. Escenario “Adicionar usuario”.

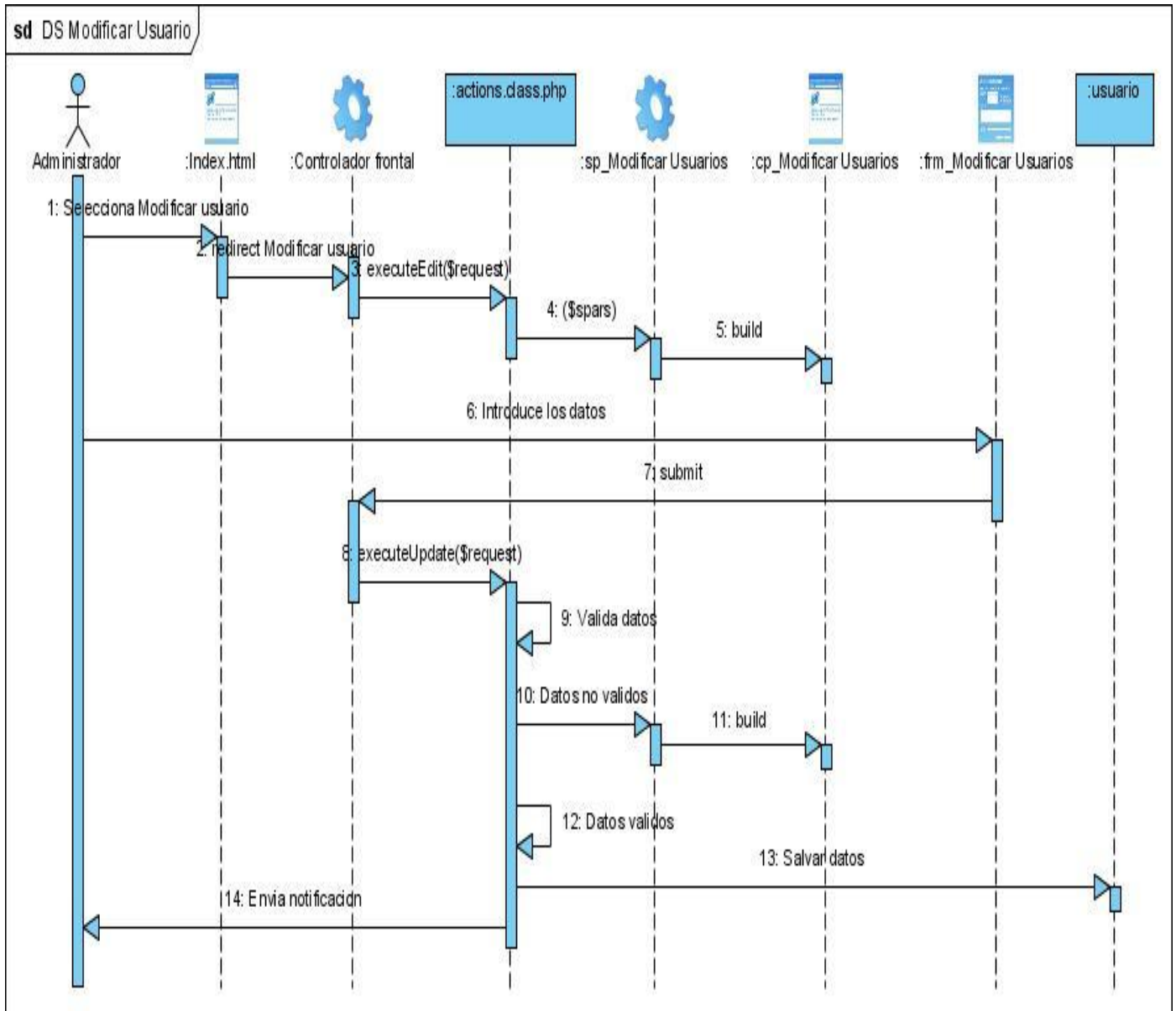


Diagrama de secuencia CU Gestionar usuario. Escenario “Modificar usuario”.

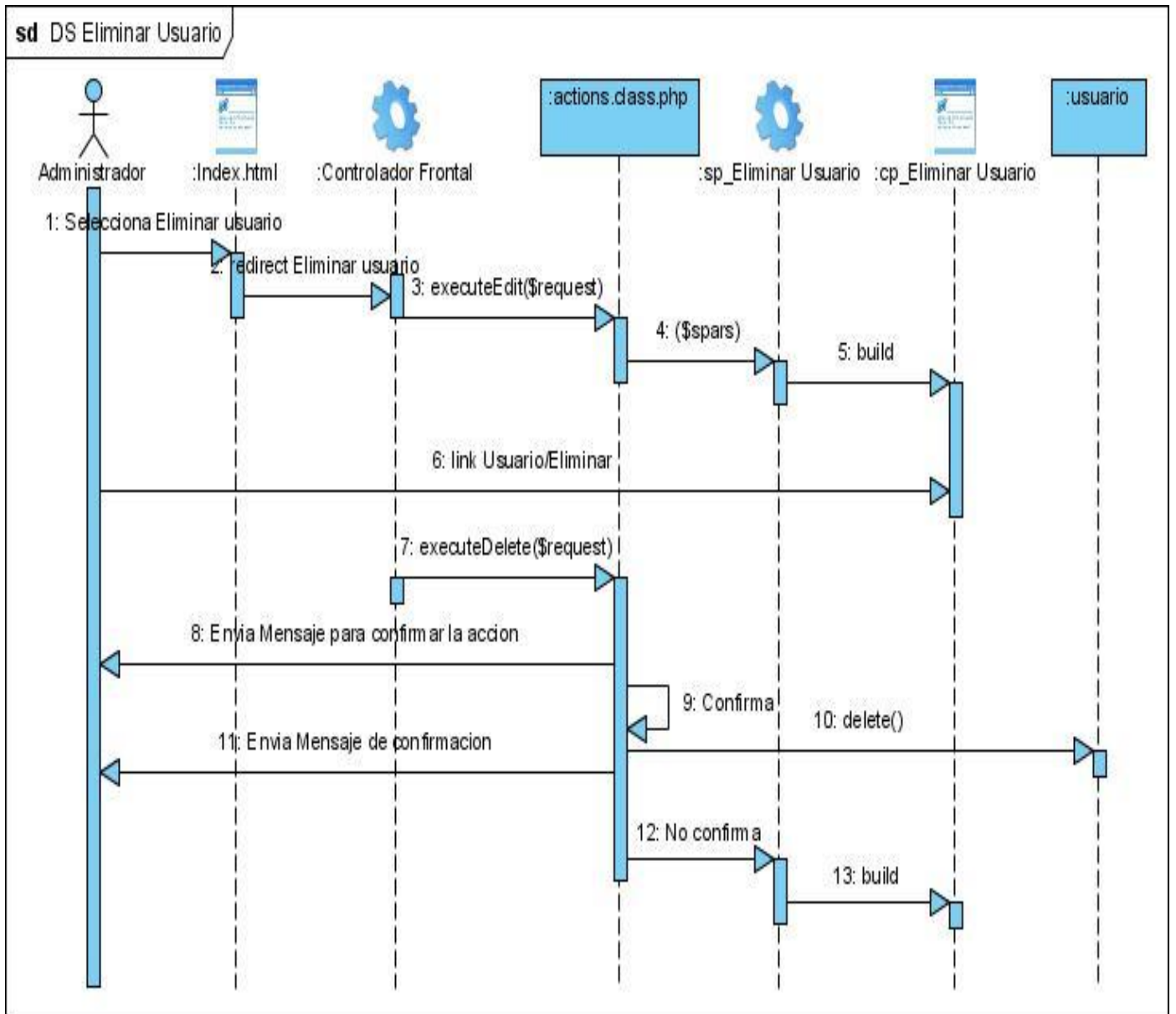


Diagrama de secuencia CU Gestionar usuario. Escenario "Eliminar usuario".

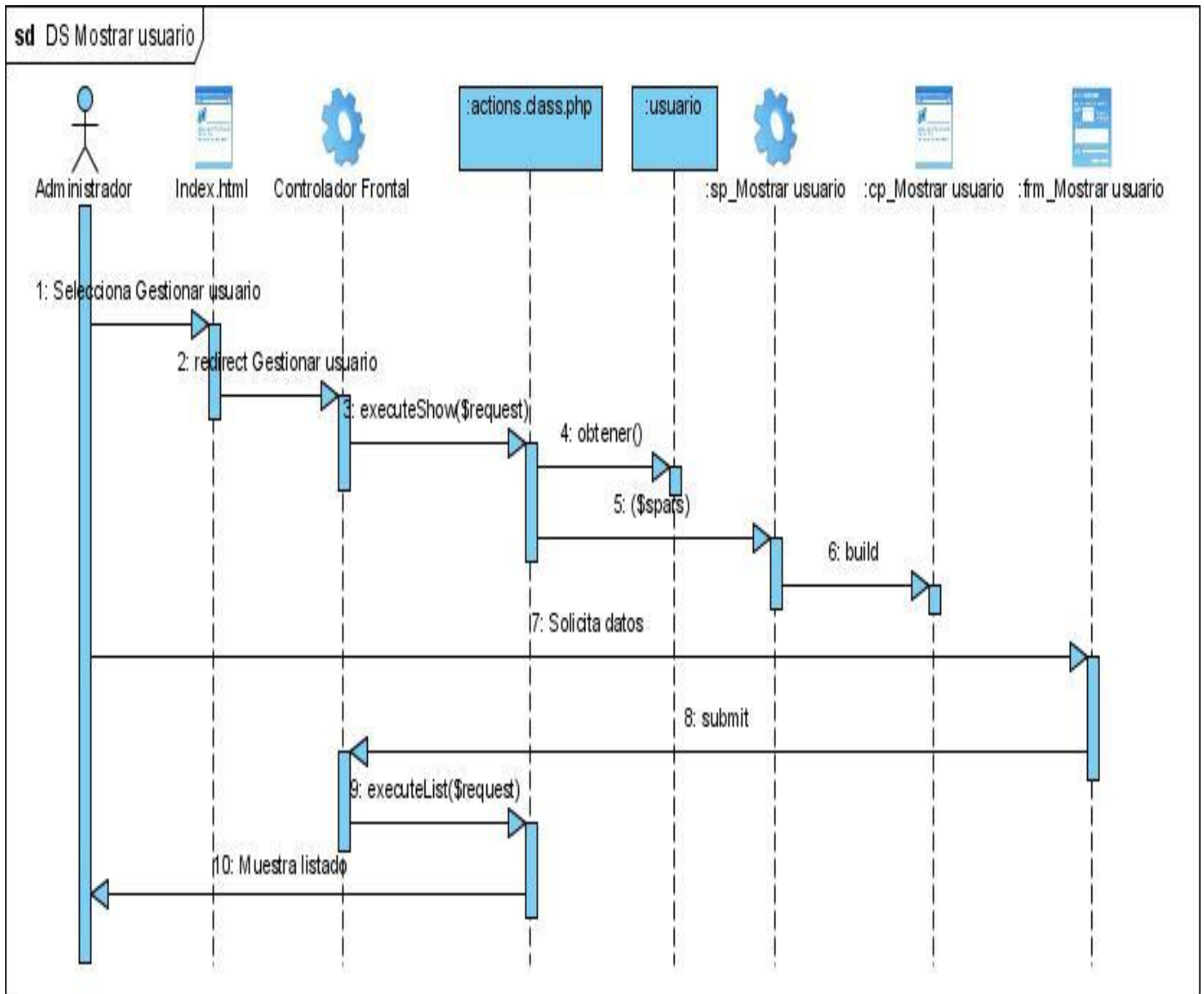


Diagrama de secuencia CU Gestionar usuario. Escenario “Mostrar usuario”.

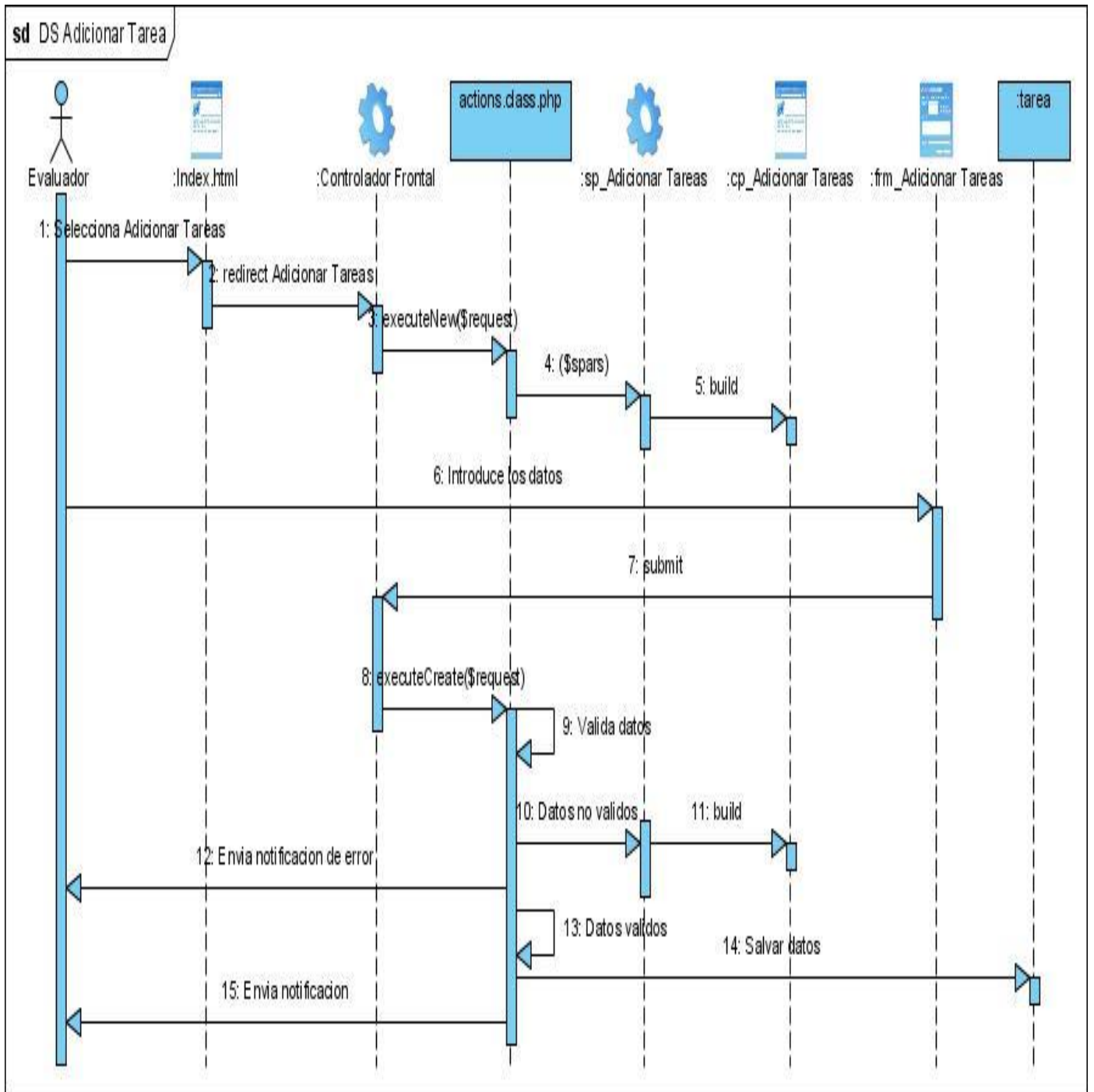


Diagrama de secuencia CU Gestionar tarea. Escenario “Adicionar tarea”.

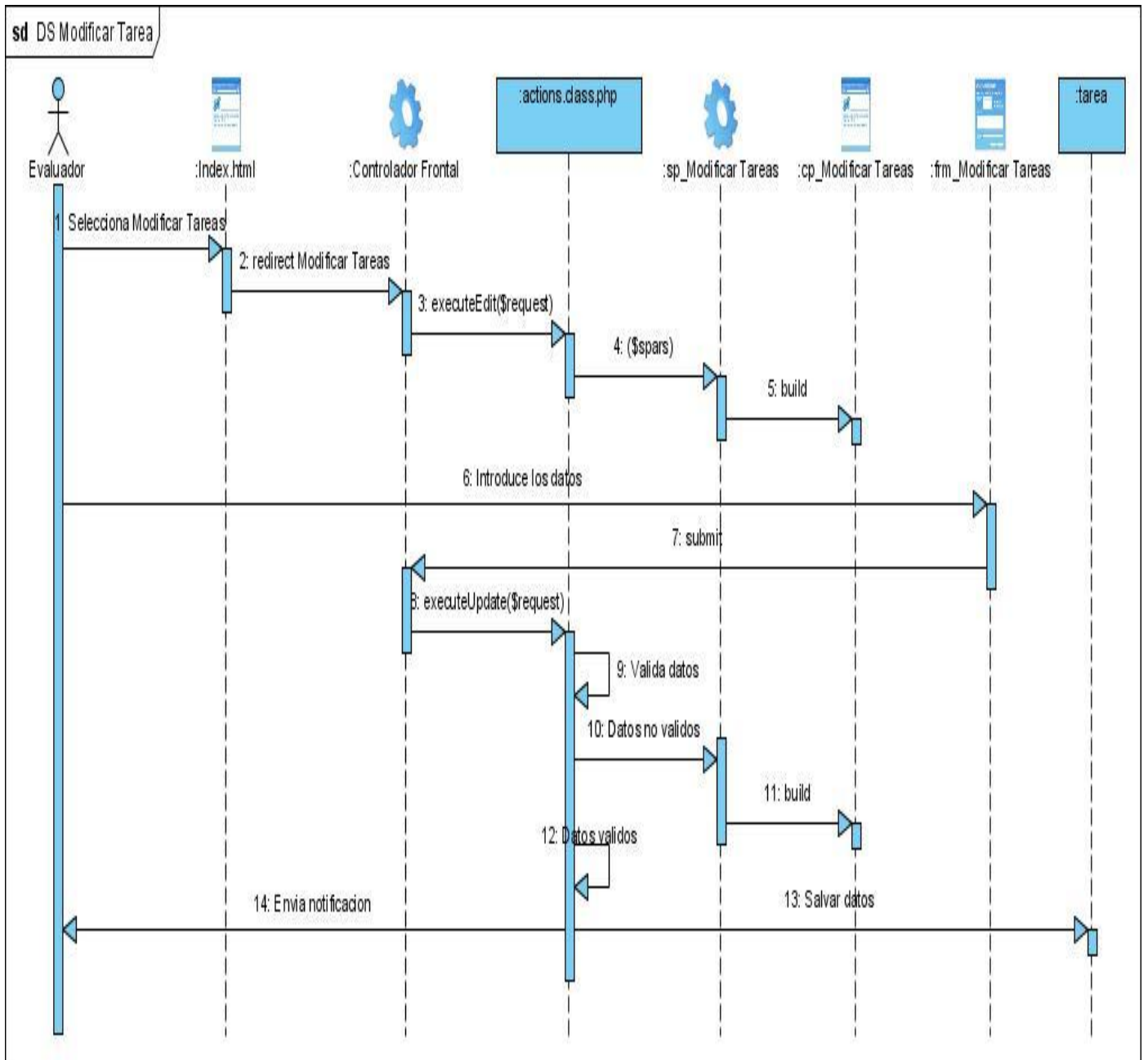


Diagrama de secuencia CU Gestionar tarea. Escenario “Modificar tarea”.

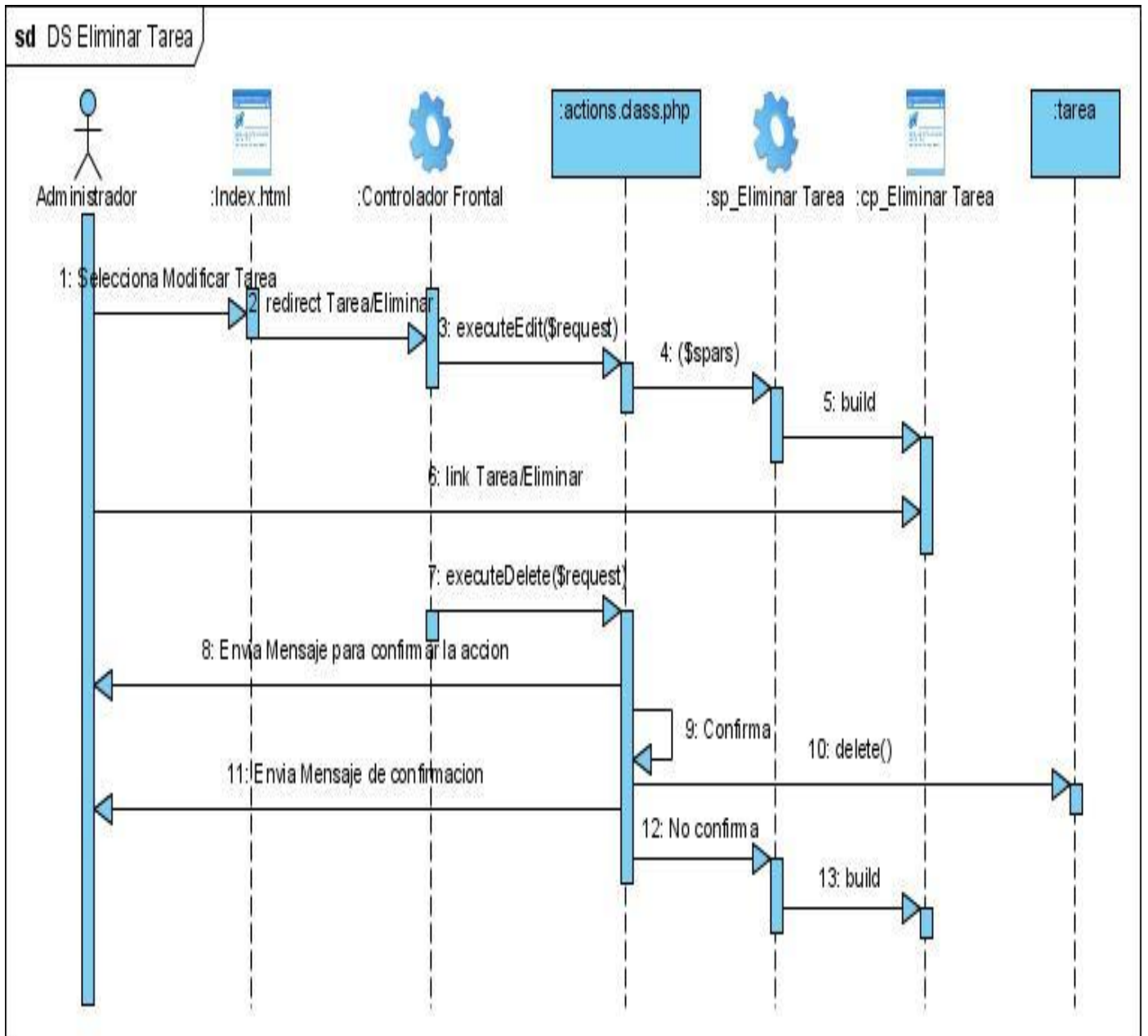


Diagrama de secuencia CU Gestionar tarea. Escenario “Eliminar tarea”.



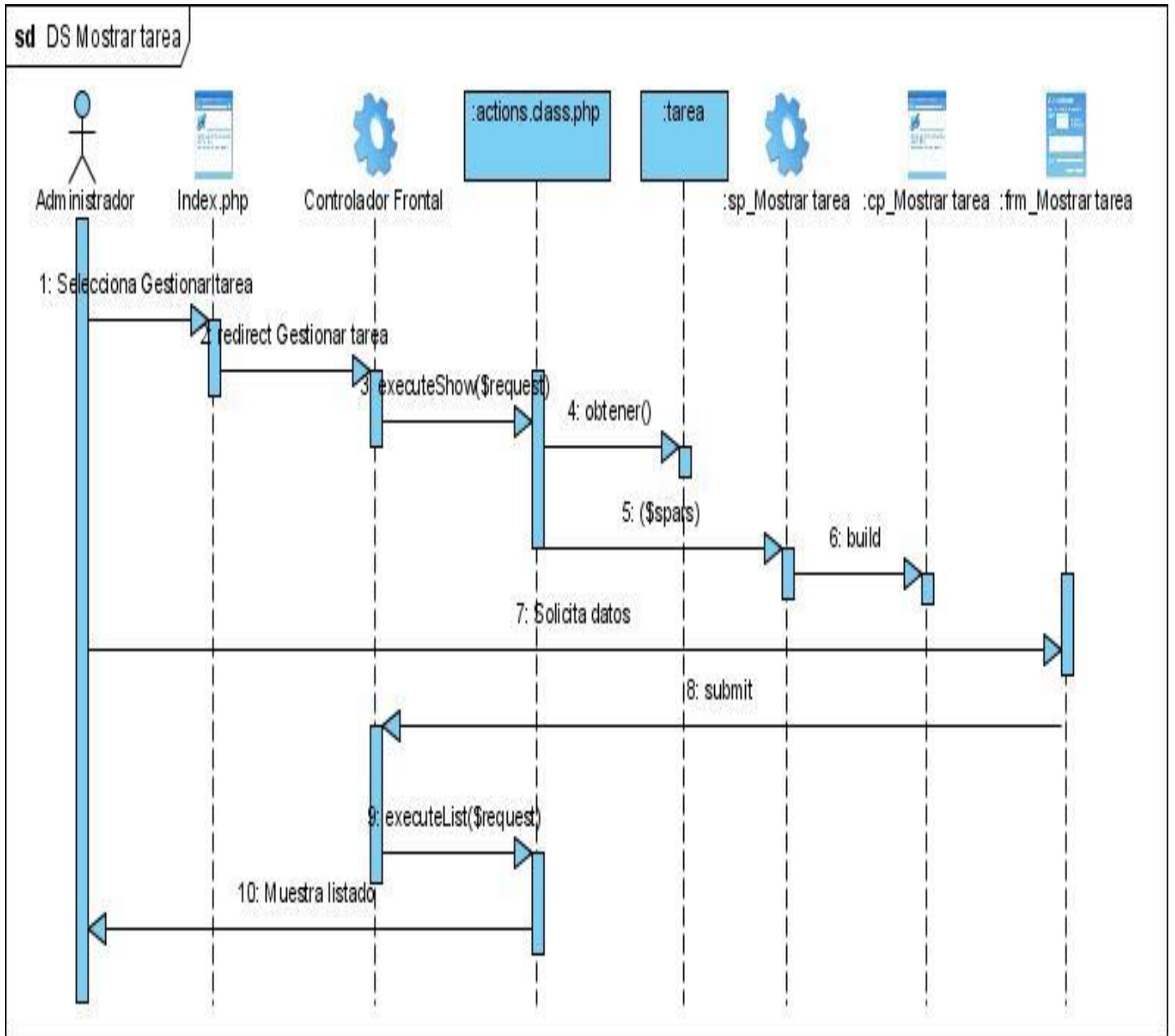


Diagrama de secuencia CU Gestionar tarea. Escenario “Mostrar tarea”.

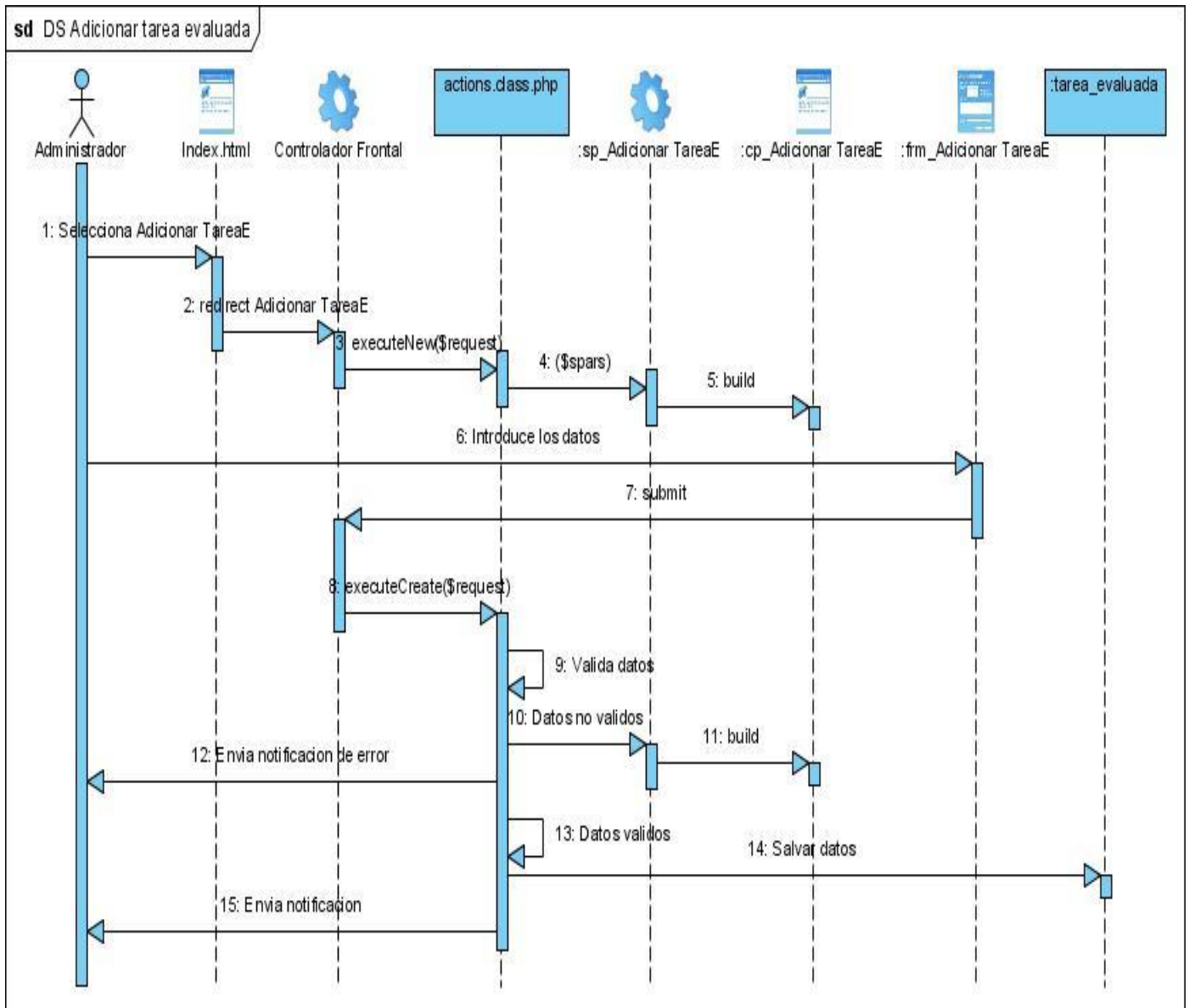


Diagrama de secuencia CU Gestionar tarea evaluada. Escenario “Adicionar tarea evaluada”.

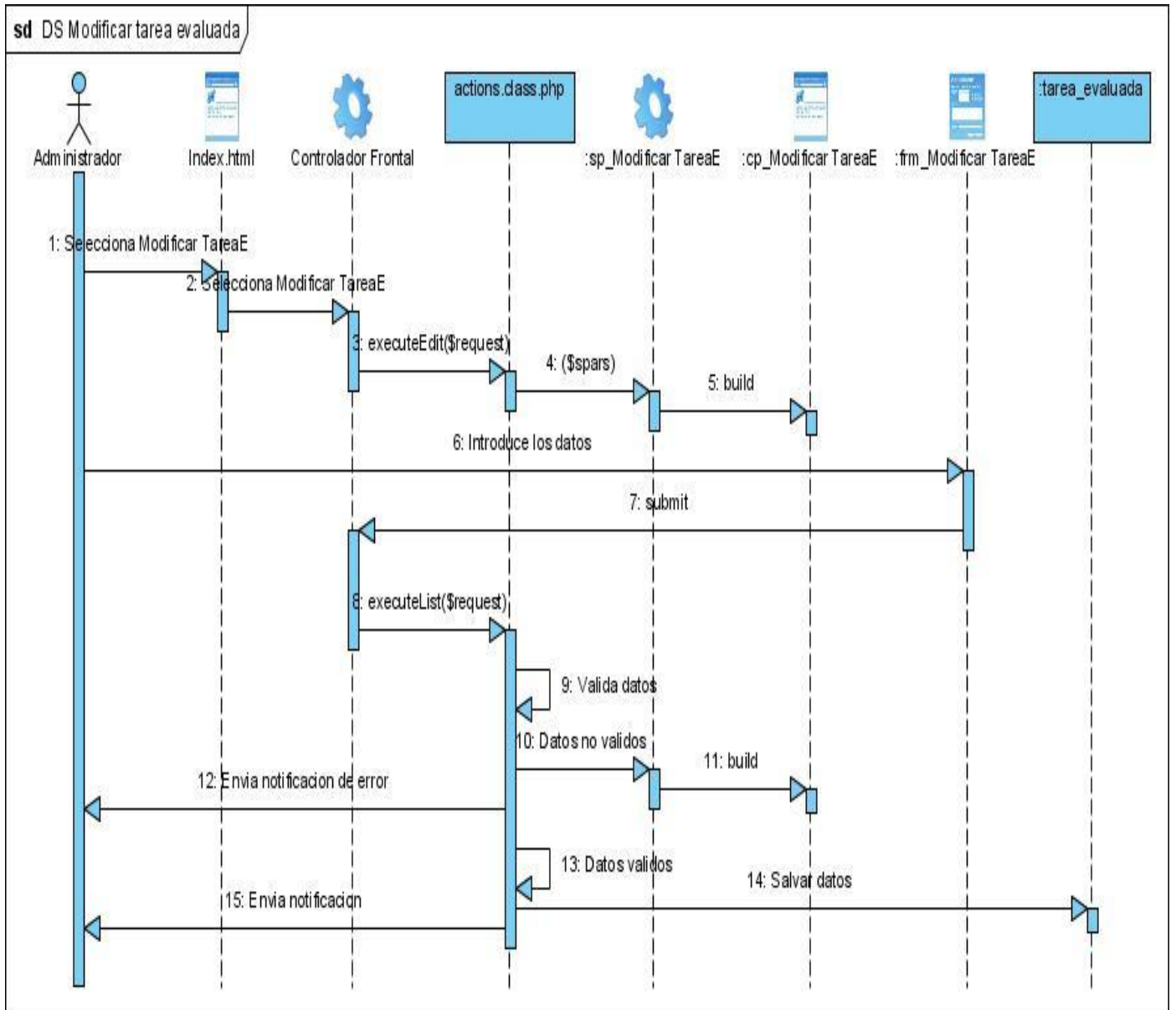


Diagrama de secuencia CU Gestionar tarea evaluada. Escenario “Modificar tarea evaluada”.

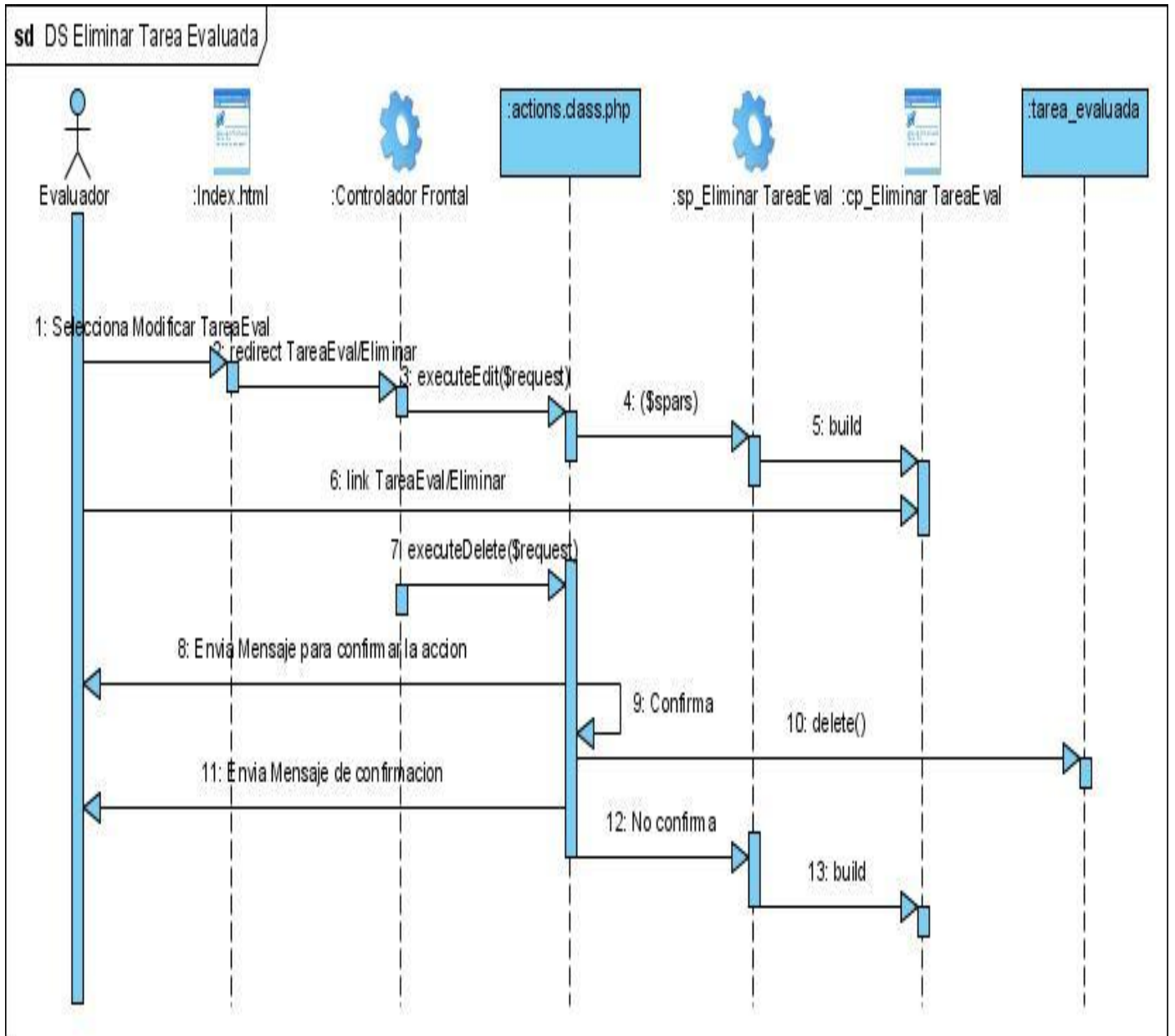


Diagrama de secuencia CU Gestionar tarea evaluada. Escenario “Eliminar tarea evaluada”.

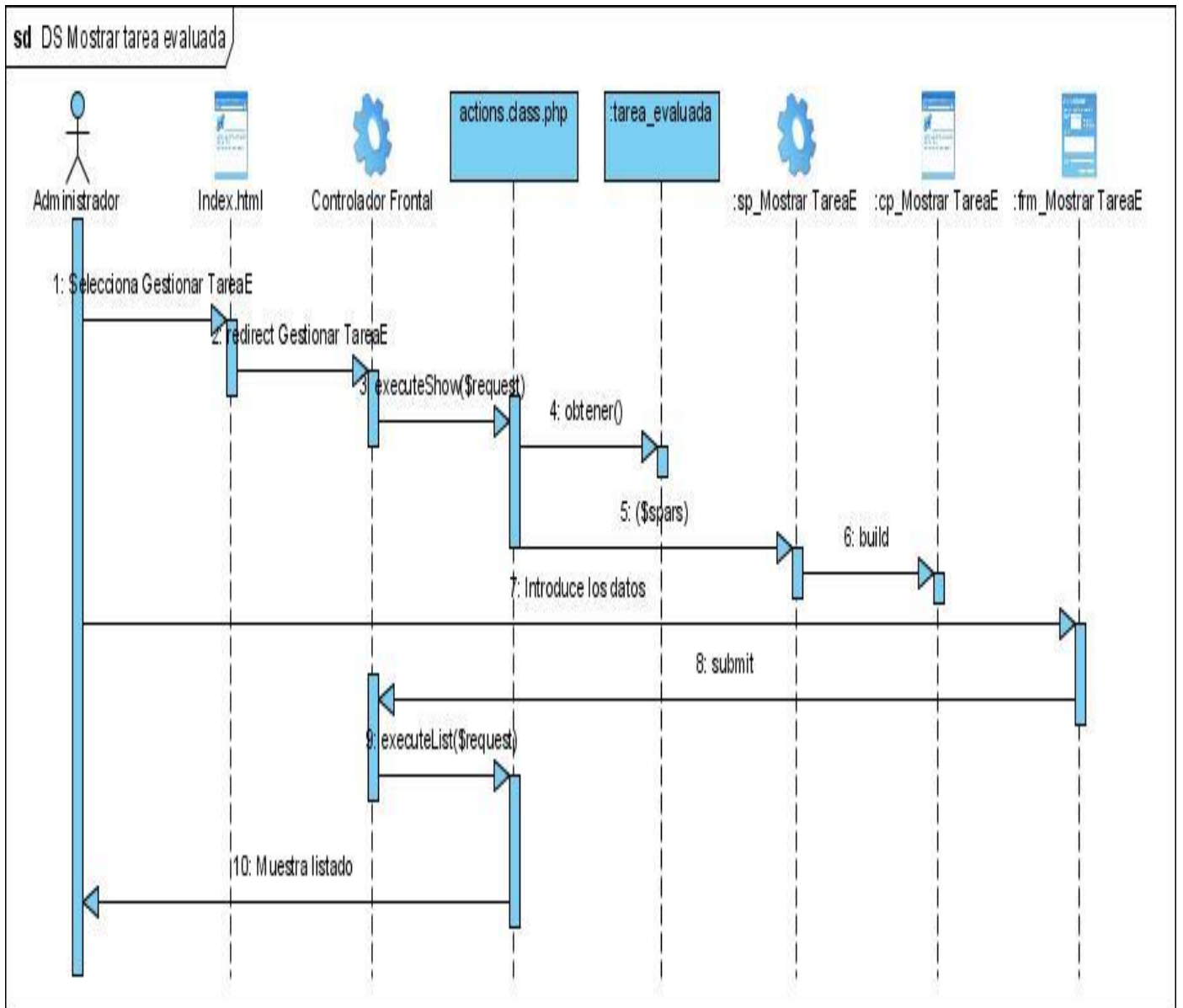


Diagrama de secuencia CU Gestionar tarea evaluada. Escenario “Mostrar tarea evaluada”.

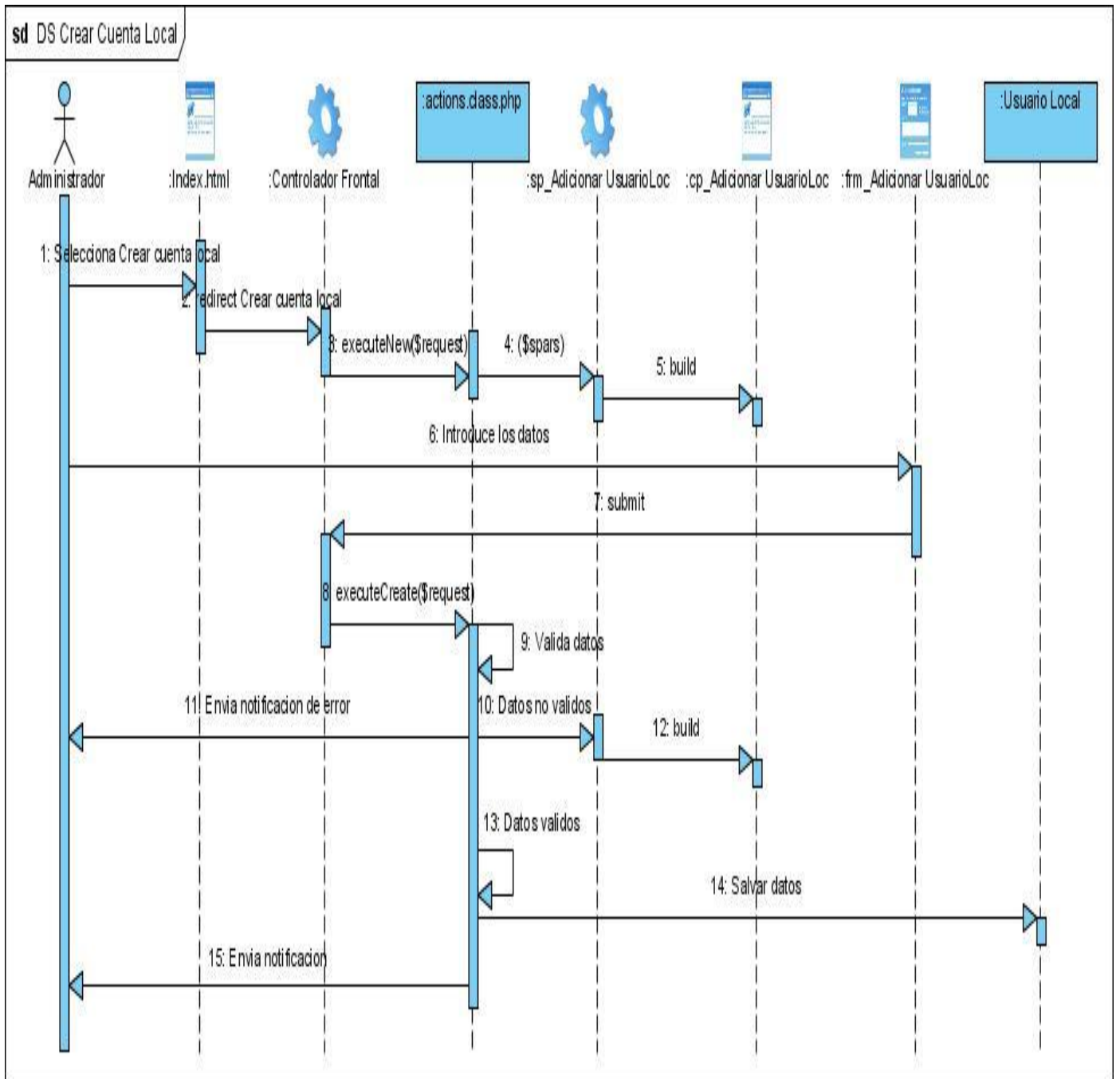


Diagrama de secuencia CU Crear cuenta local.

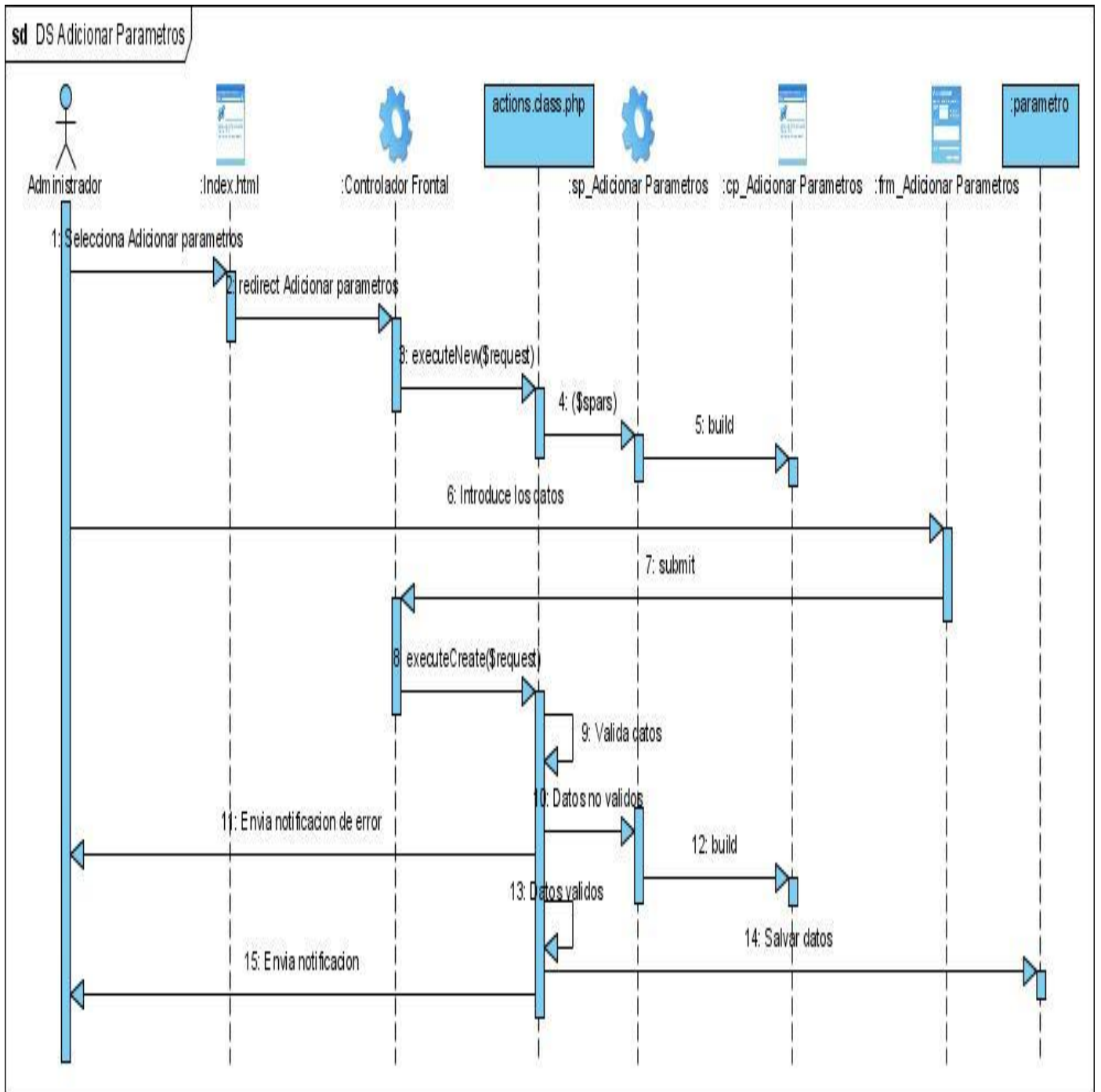


Diagrama de secuencia CU Gestionar parámetro. Escenario “Adicionar parámetro”.

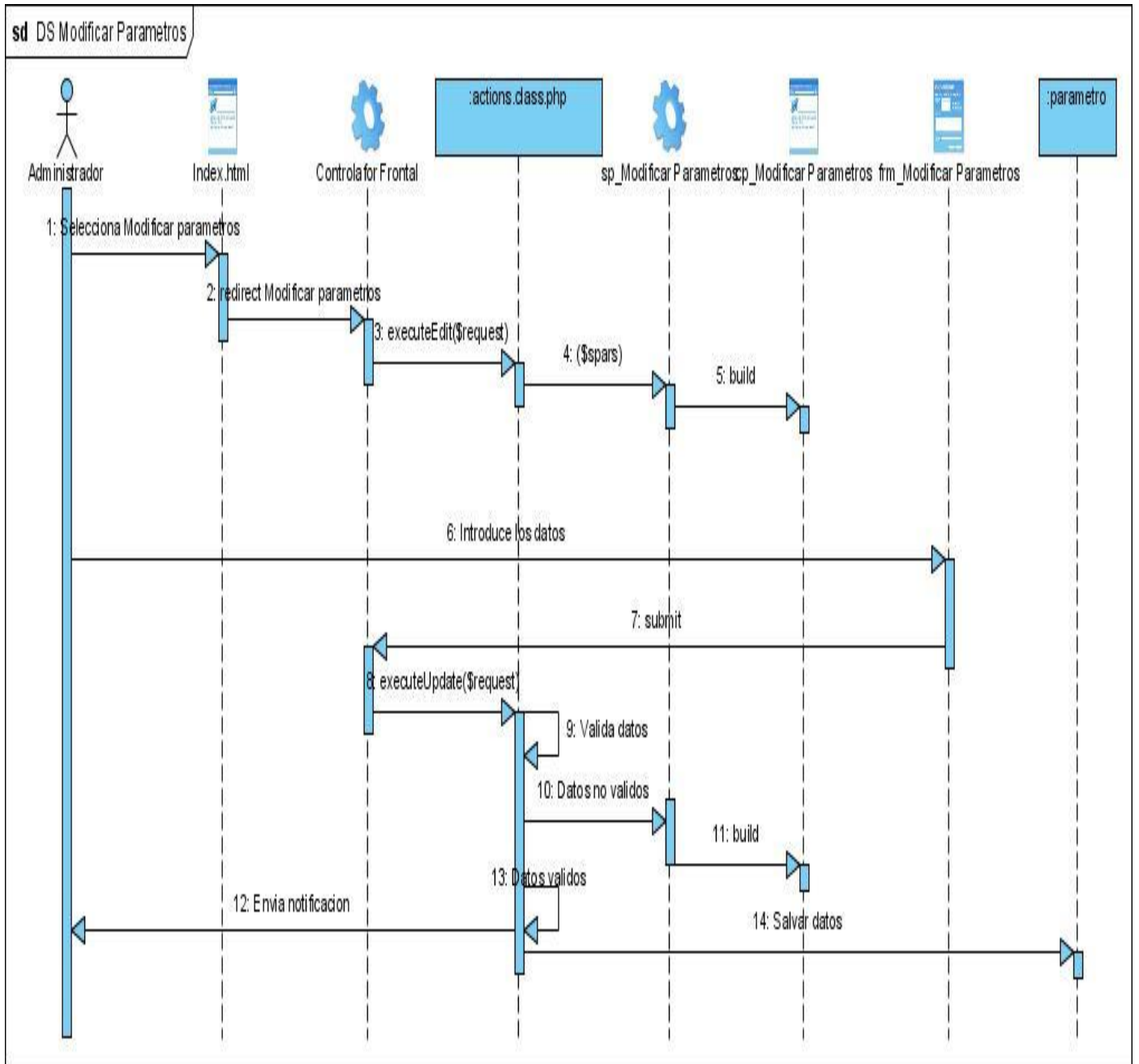


Diagrama de secuencia CU Gestionar parámetro. Escenario “Modificar parámetro”.



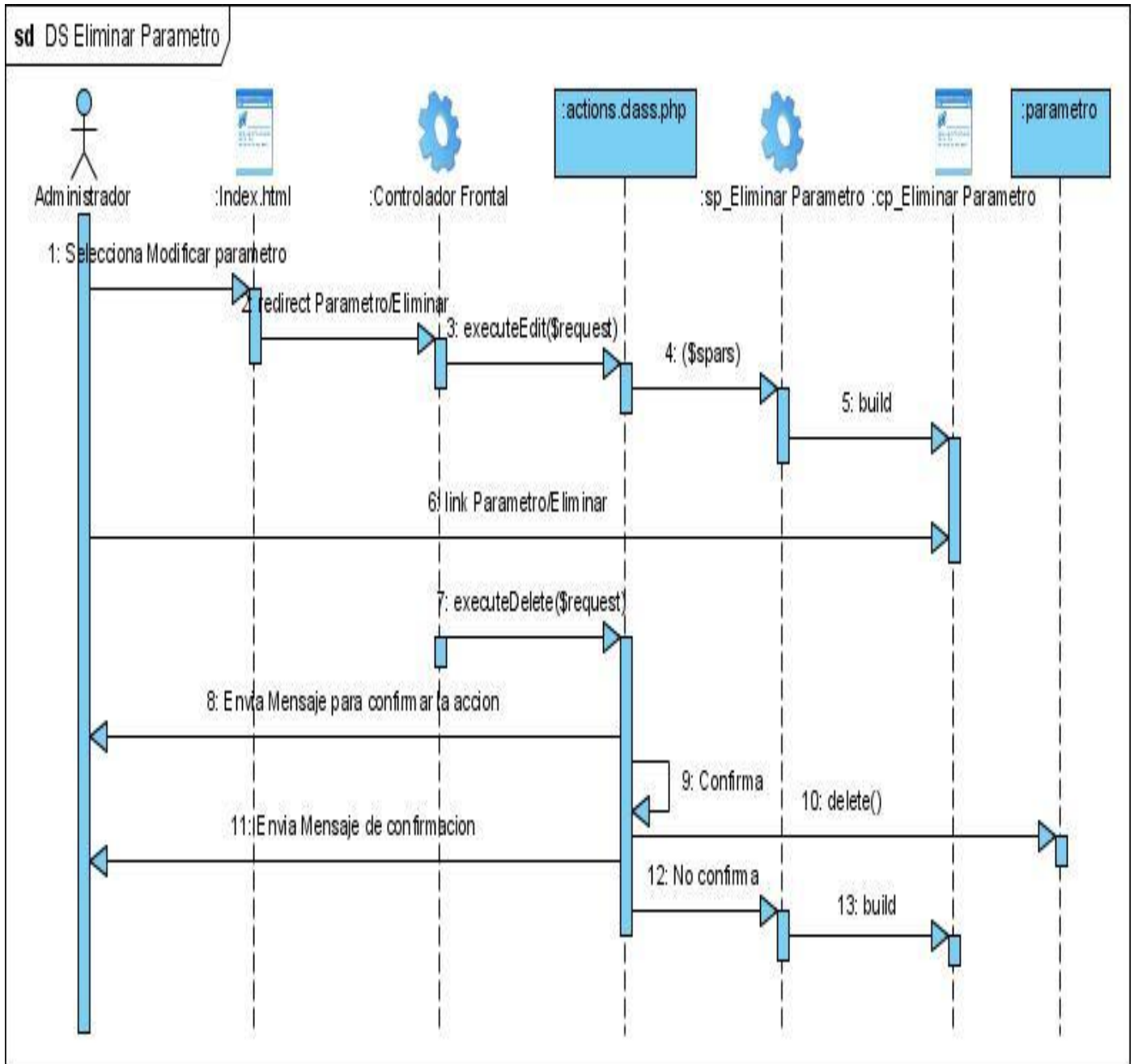


Diagrama de secuencia CU Gestionar parámetro. Escenario “Eliminar parámetro”.

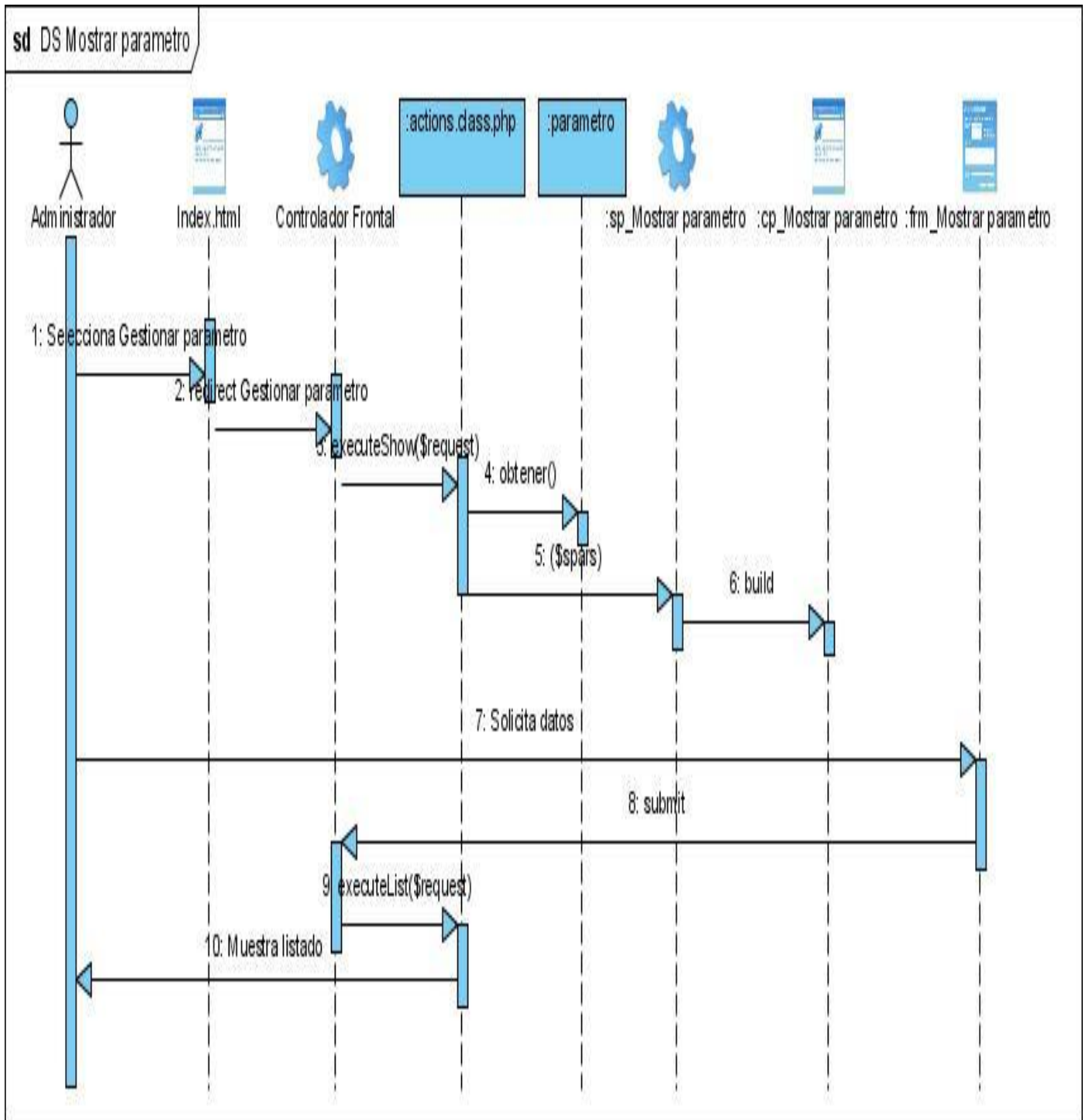


Diagrama de secuencia CU Gestionar parámetro. Escenario “Mostrar parámetro”.

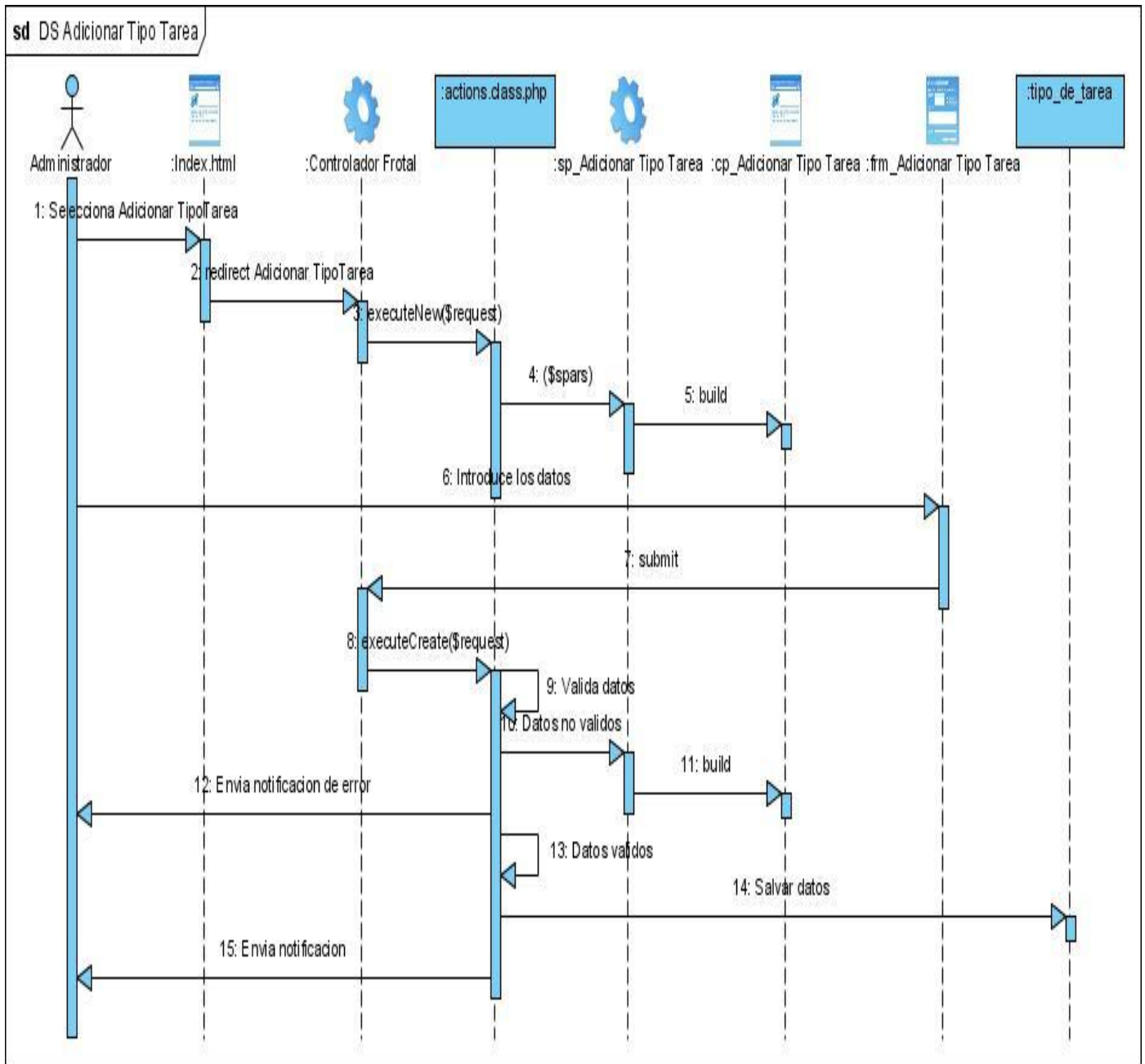


Diagrama de secuencia CU Gestionar tipo de tarea. Escenario “Adicionar tipo de tarea”.

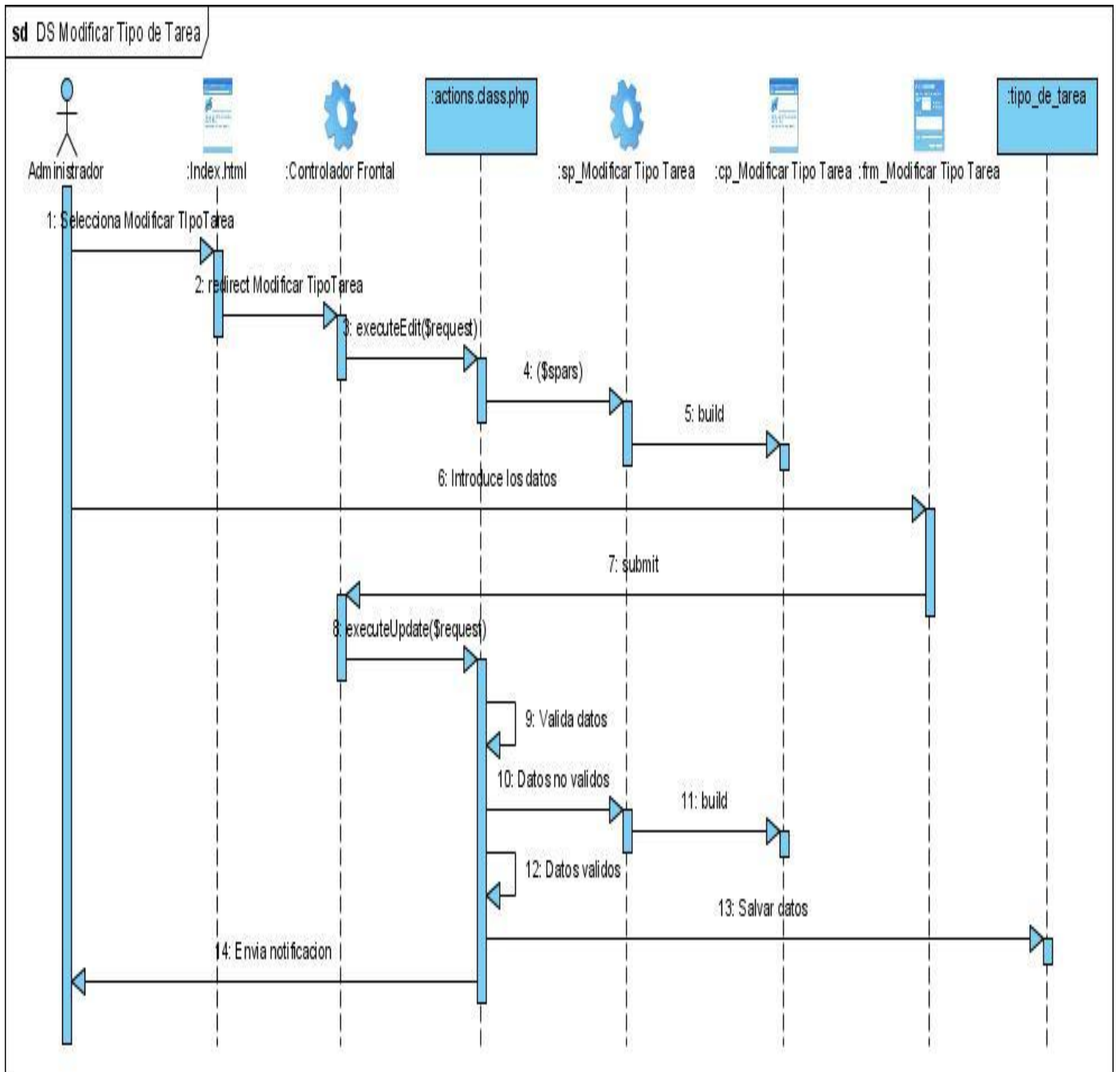


Diagrama de secuencia CU Gestionar tipo de tarea. Escenario “Modificar tipo de tarea”.

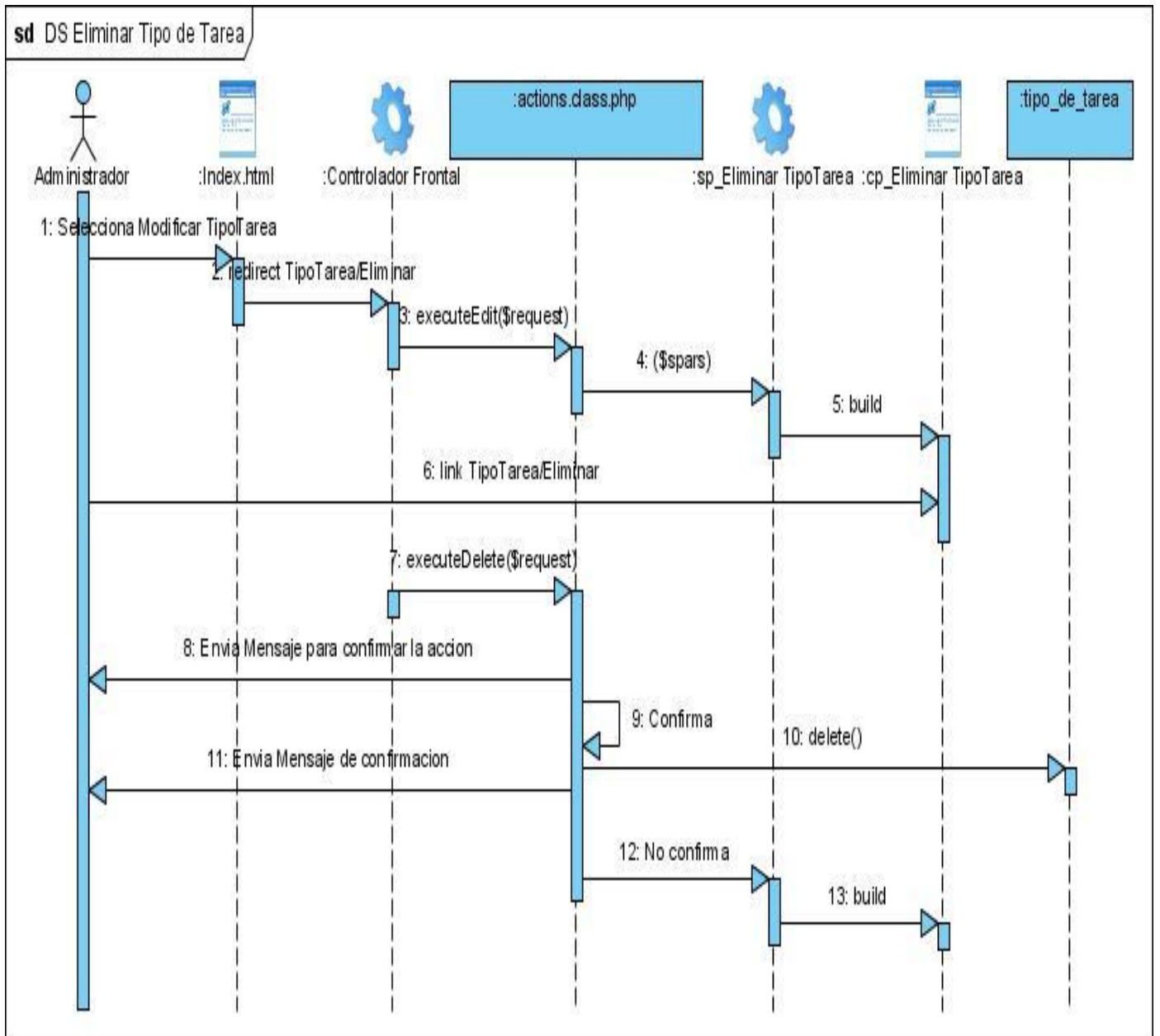


Diagrama de secuencia CU Gestionar tipo de tarea. Escenario “Eliminar tipo de tarea”.

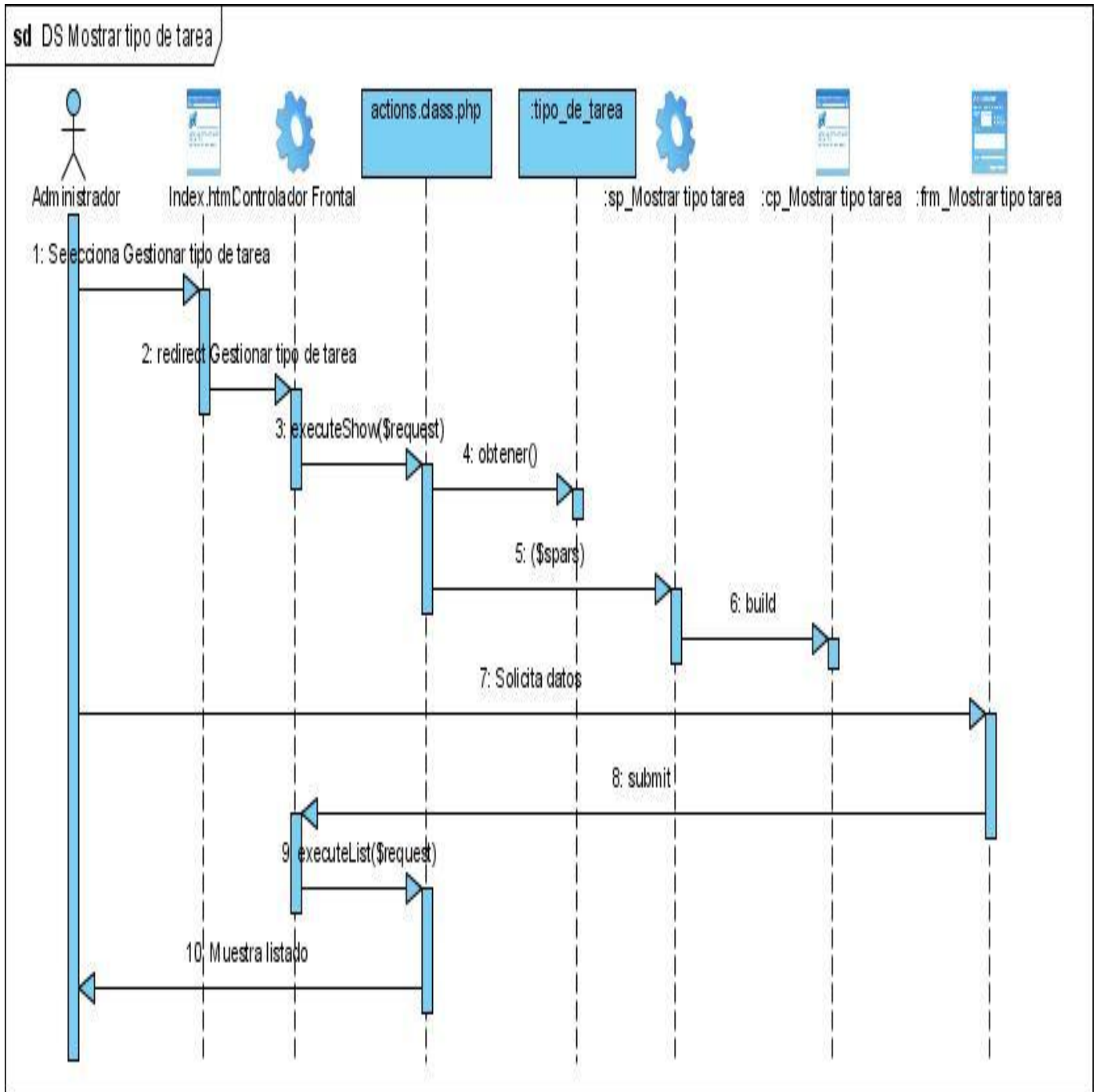


Diagrama de secuencia CU Gestionar tipo de tarea. Escenario “Mostrar tipo de tarea”.

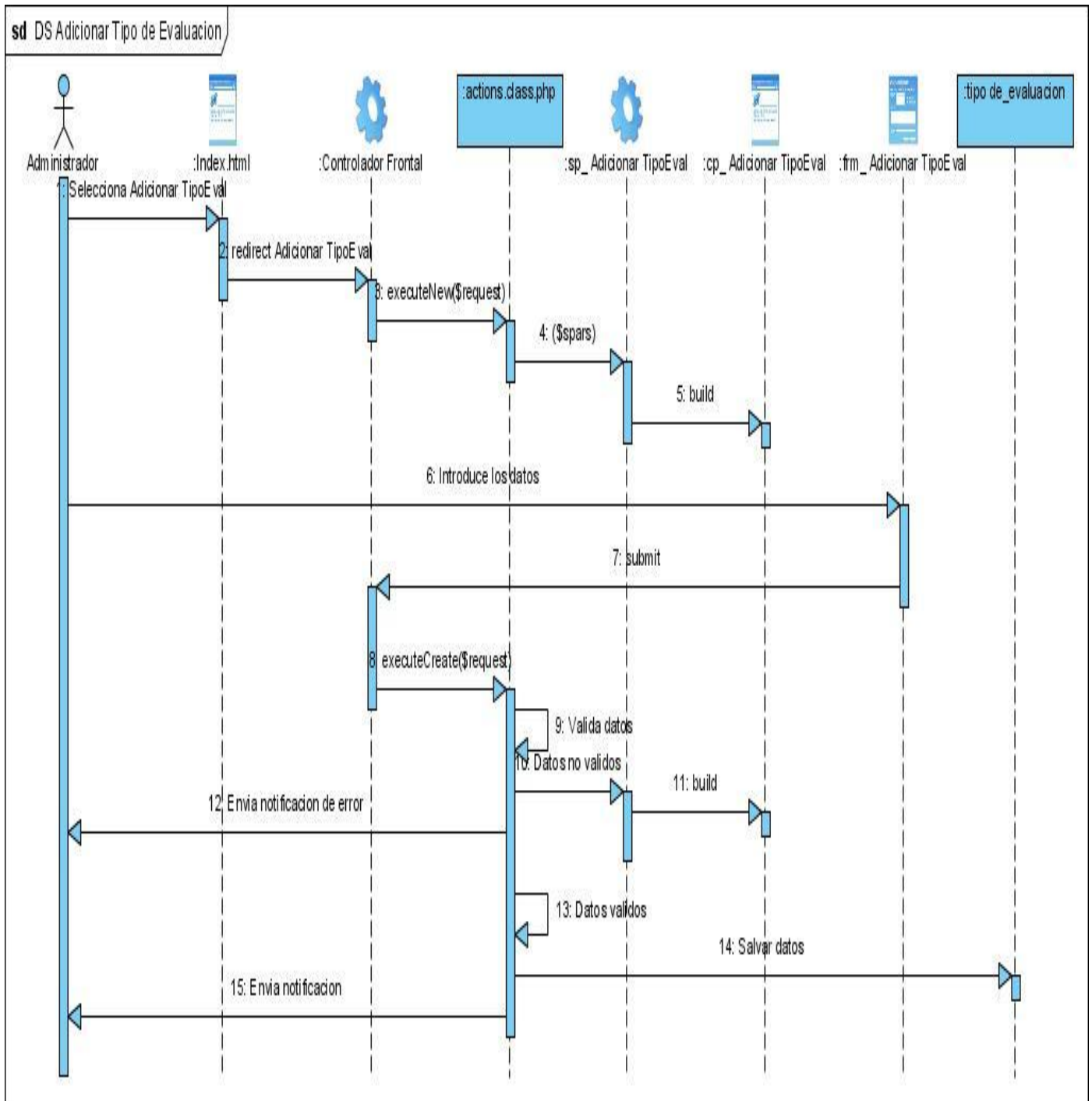


Diagrama de secuencia CU Gestionar tipo de evaluación. Escenario “Adicionar tipo de evaluación”.



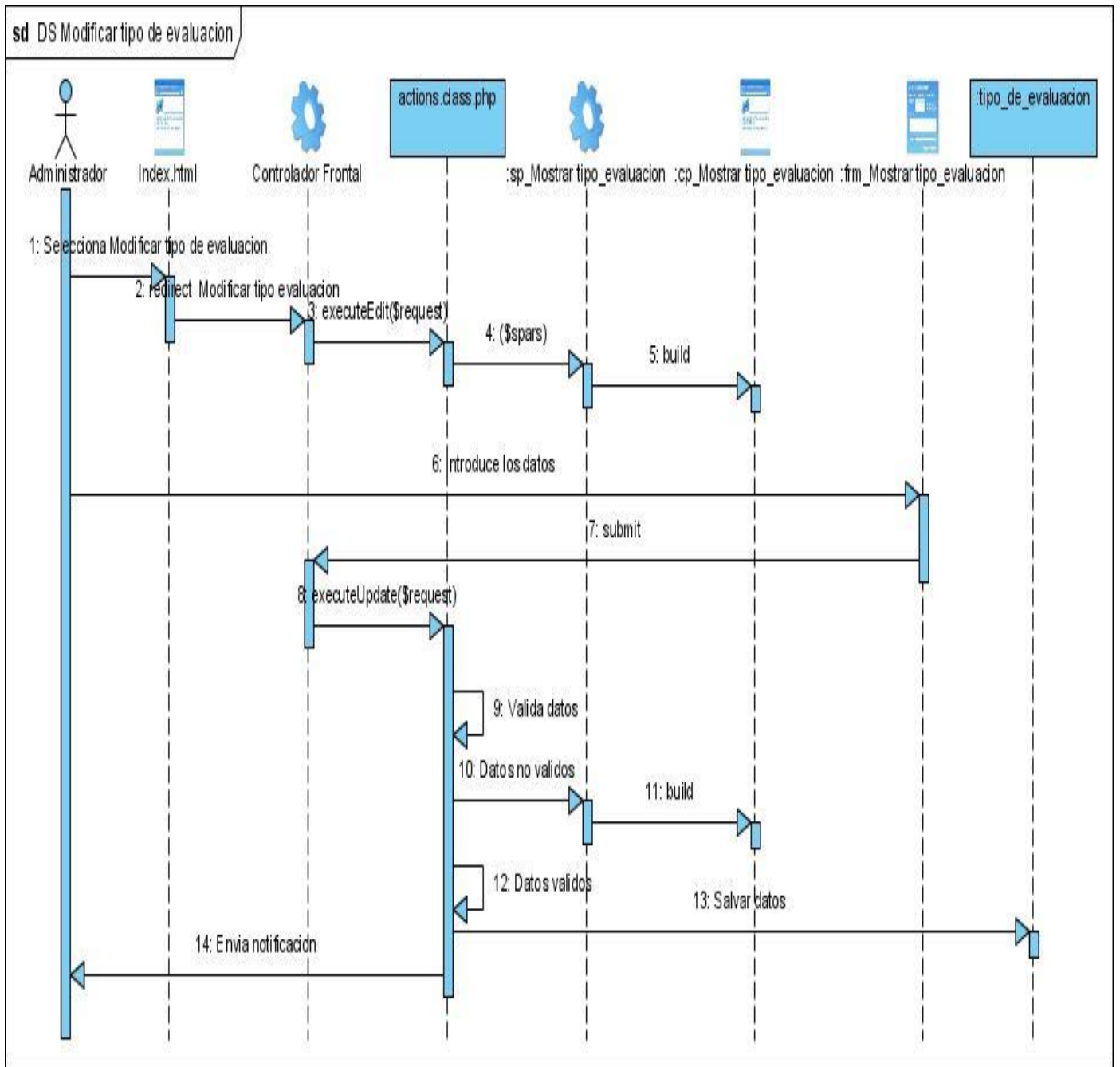


Diagrama de secuencia CU Gestionar tipo de evaluación. Escenario “Modificar tipo de evaluación”.



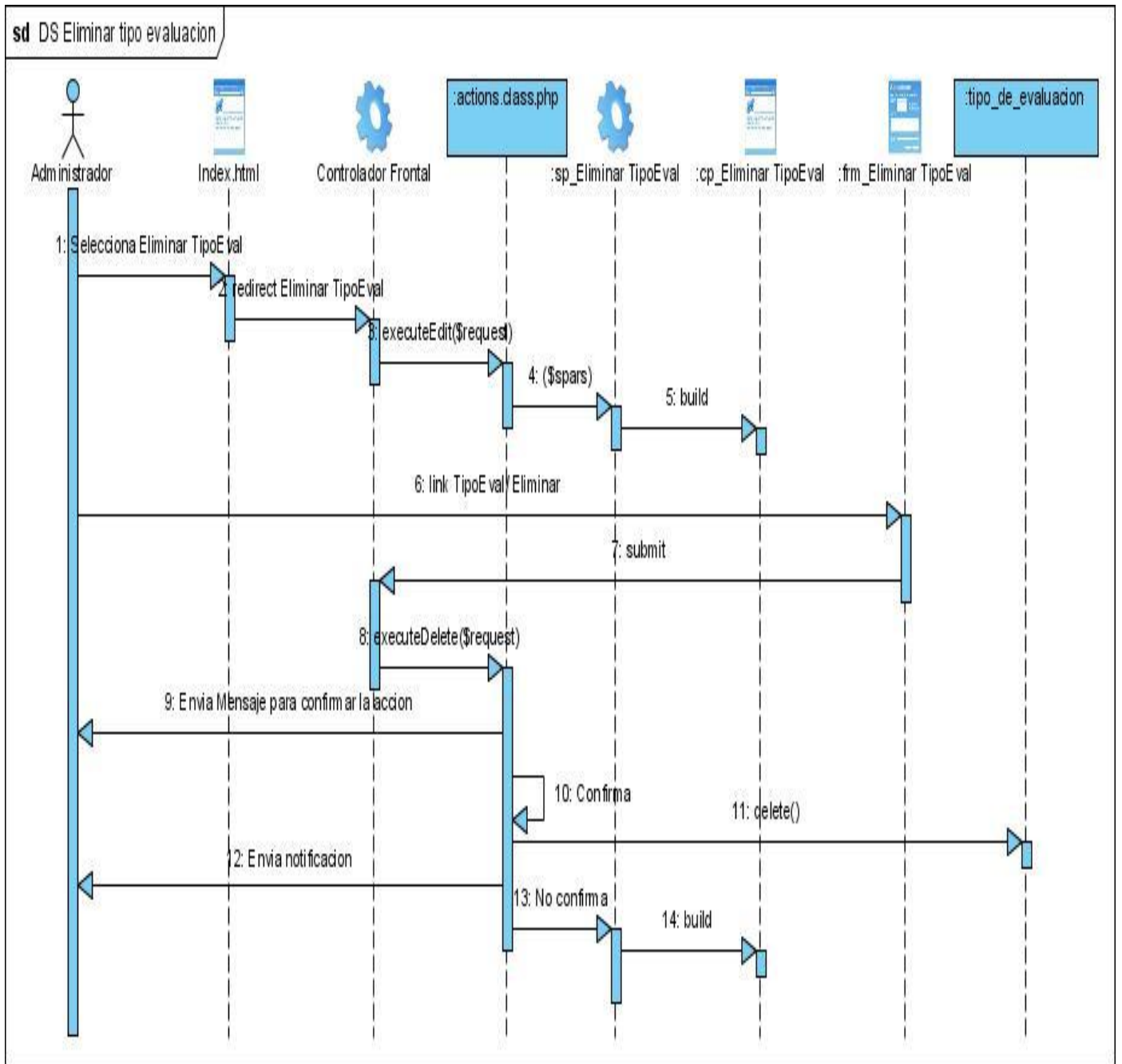


Diagrama de secuencia CU Gestionar tipo de evaluación. Escenario “Eliminar tipo de evaluación”.

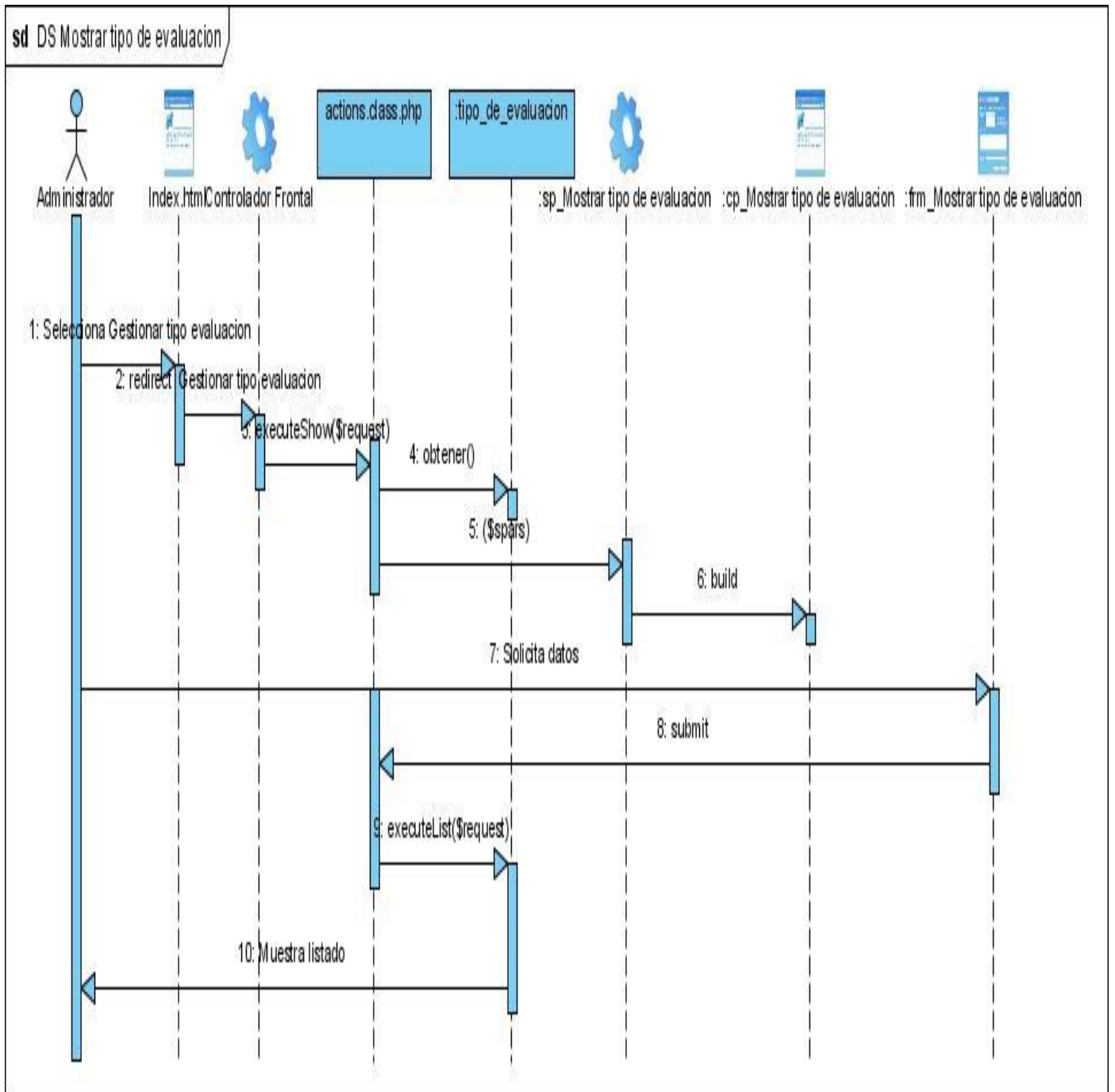


Diagrama de secuencia CU Gestionar tipo de evaluación. Escenario “Mostrar tipo de evaluación”.

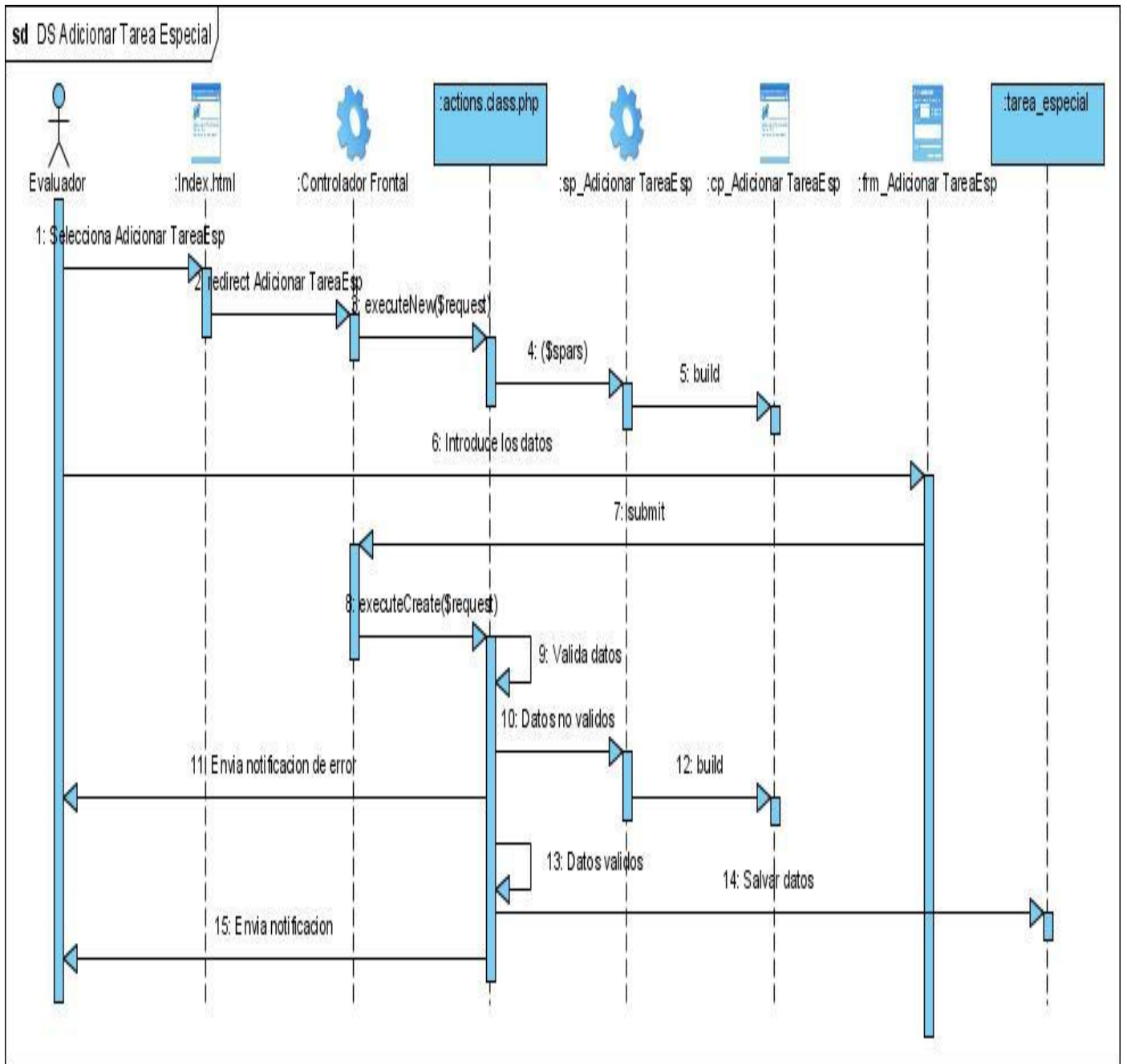


Diagrama de secuencia CU Gestionar tarea especial. Escenario “Adicionar tarea especial”.

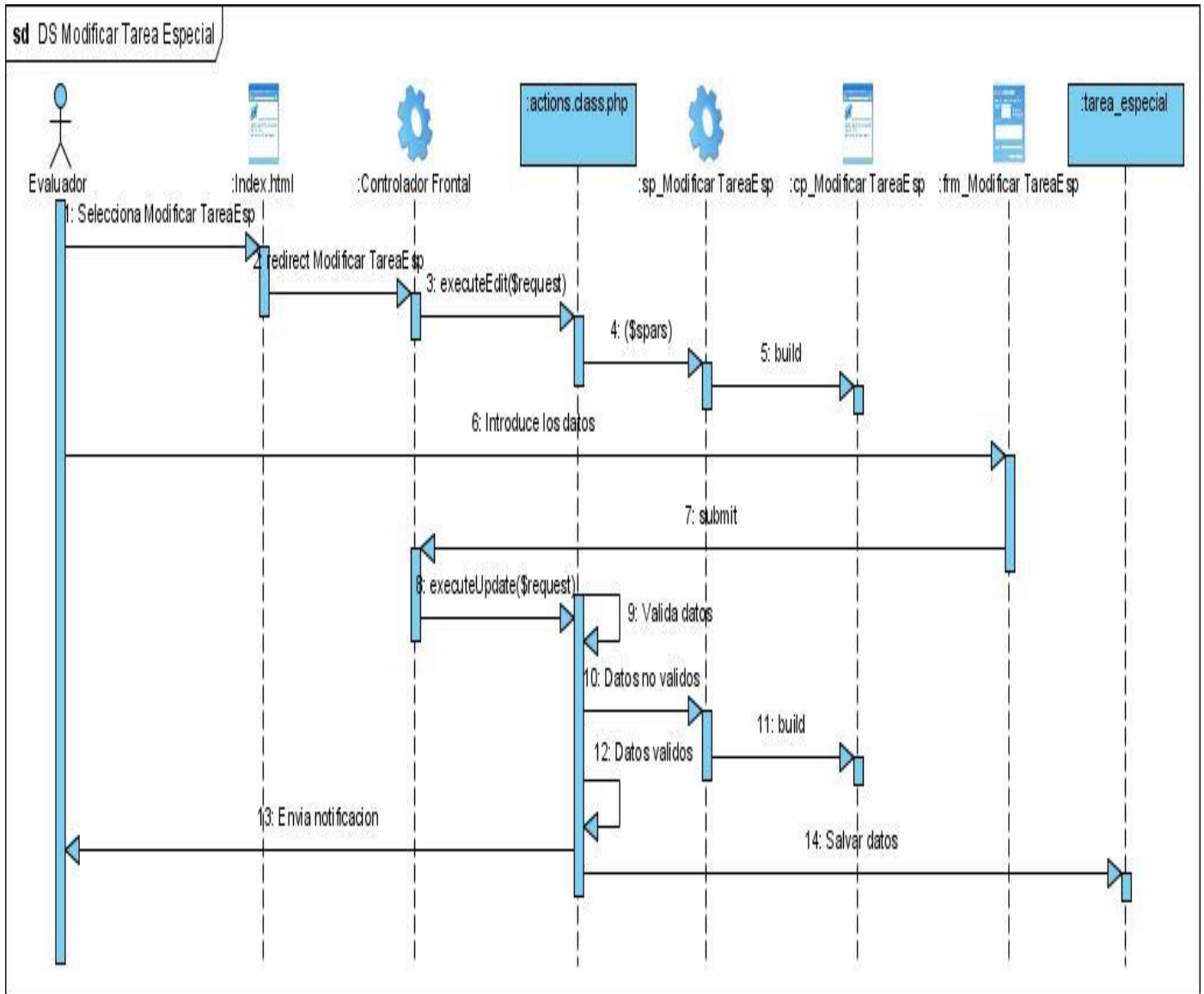


Diagrama de secuencia CU Gestionar tarea especial. Escenario “Modificar tarea especial”.

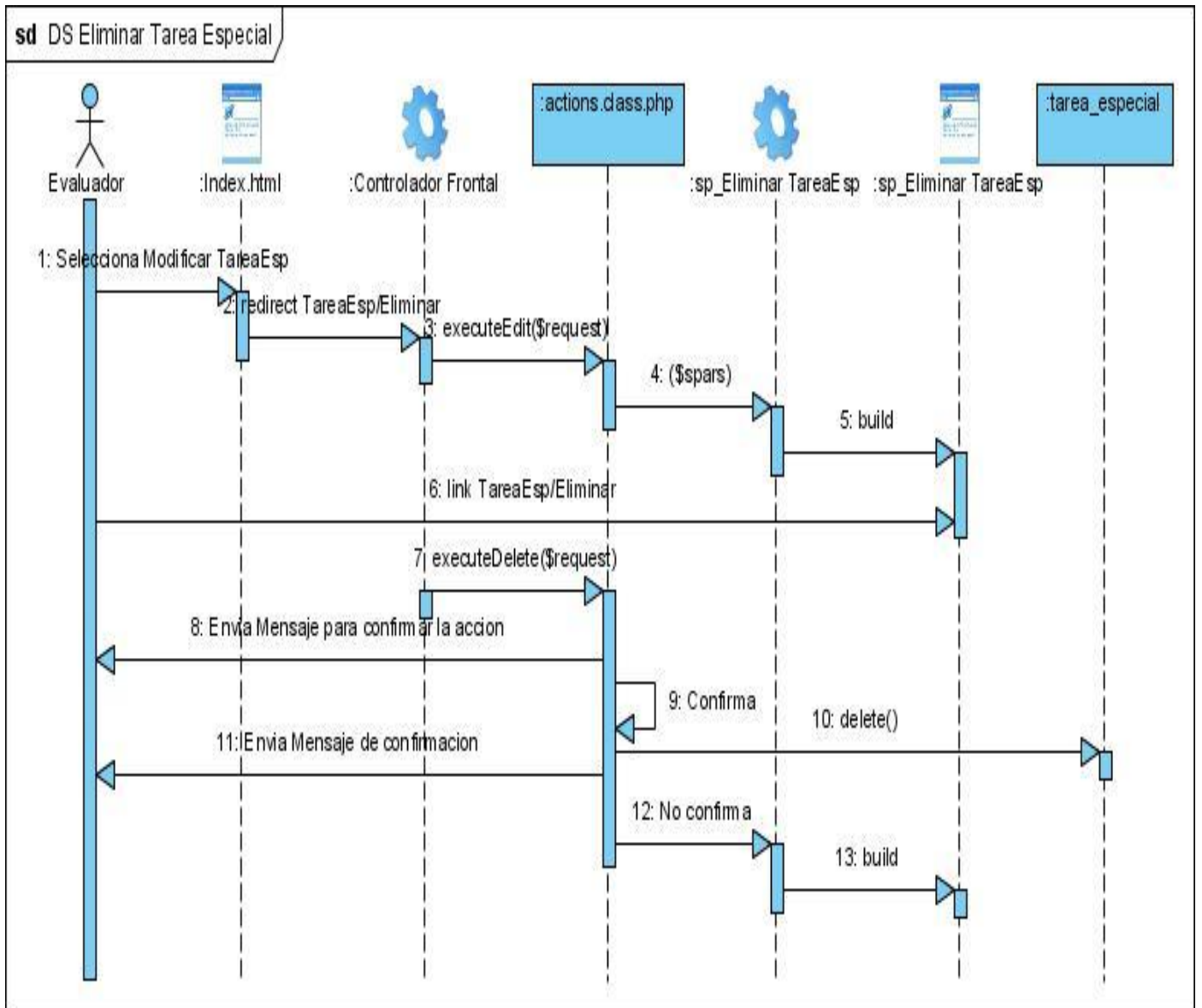


Diagrama de secuencia CU Gestionar tarea especial. Escenario “Eliminar tarea especial”.

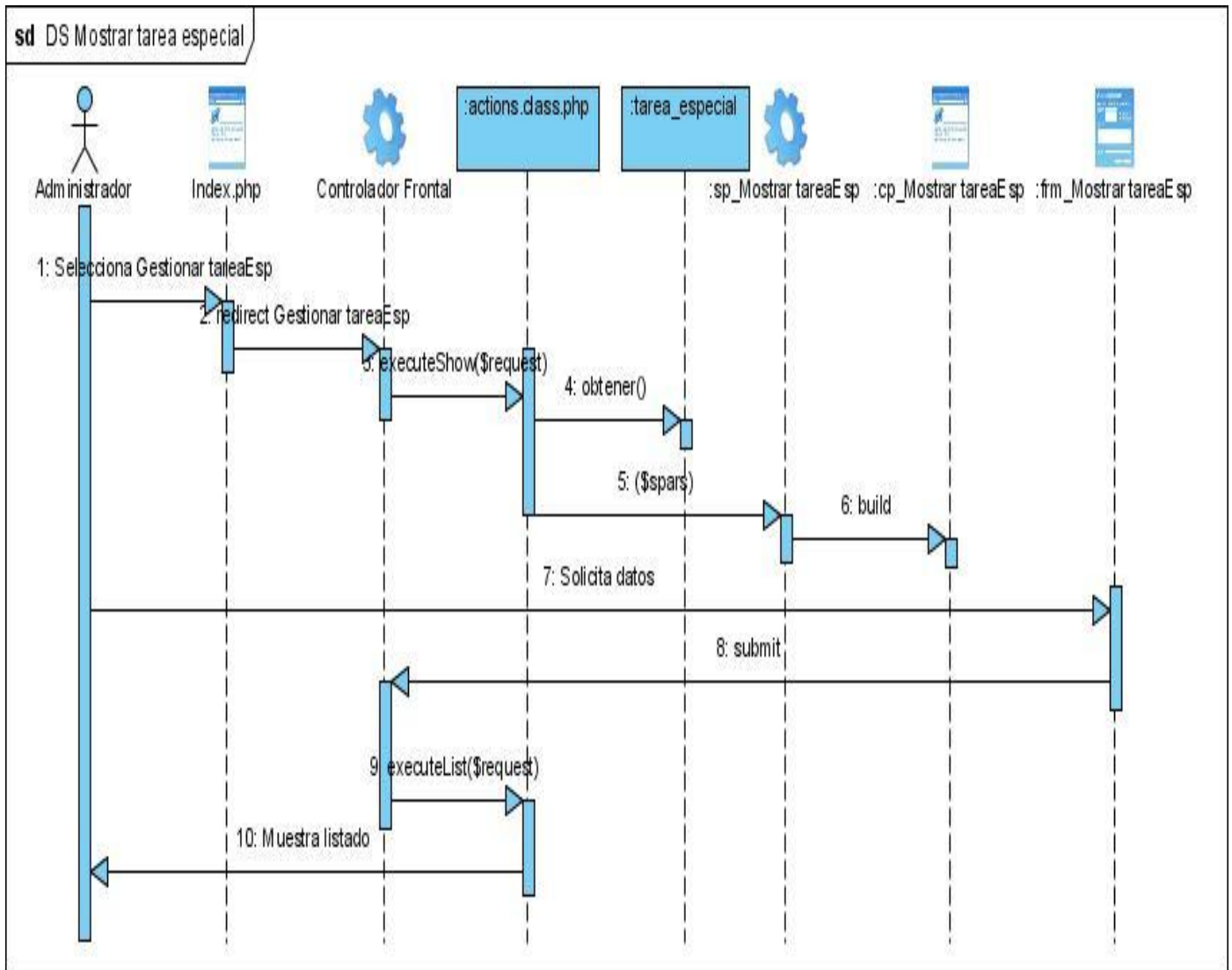


Diagrama de secuencia CU Gestionar tarea especial. Escenario “Mostrar tarea especial”.

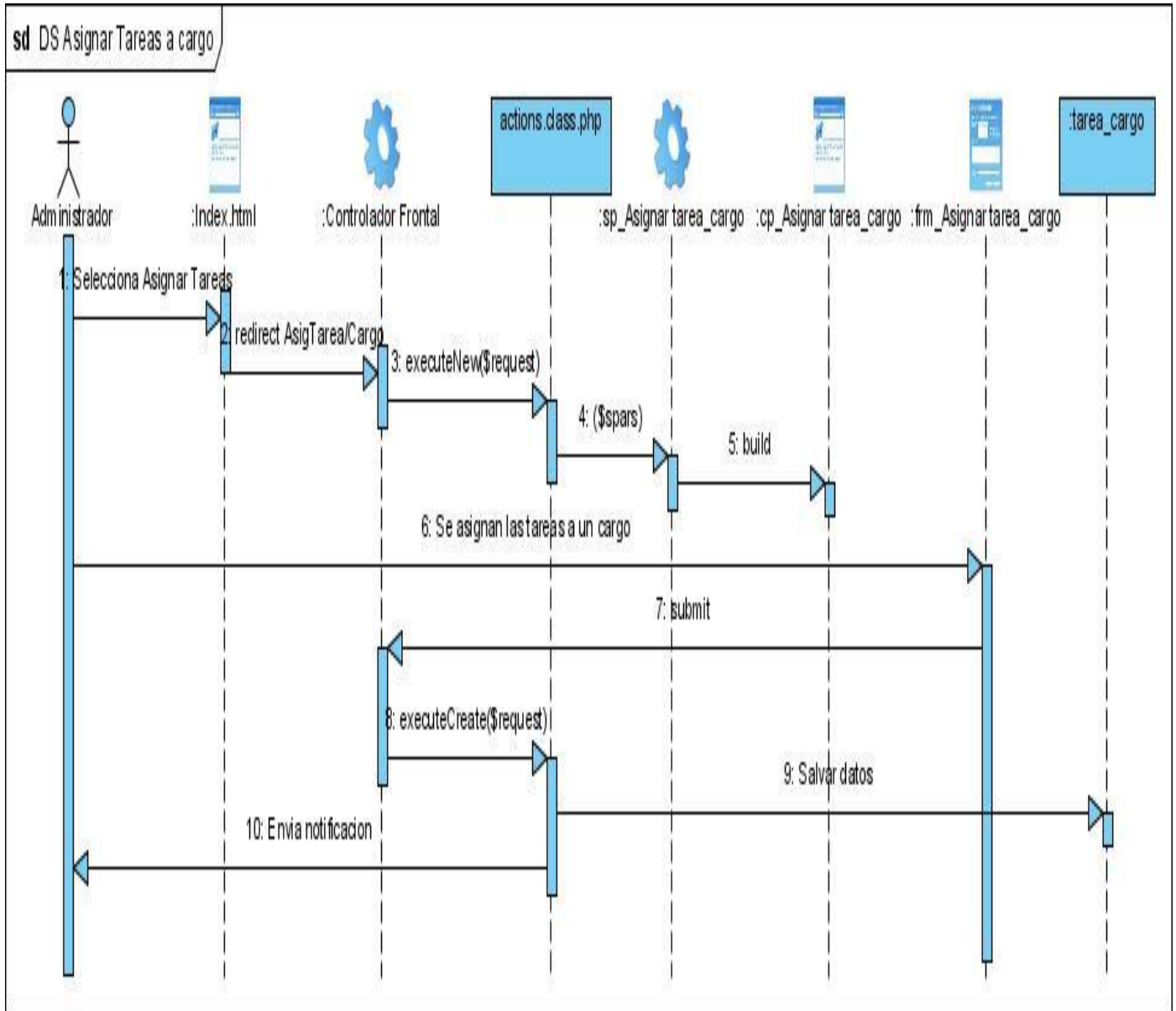


Diagrama de secuencia CU Asignar tarea. Escenario “Asignar tarea a cargo”.

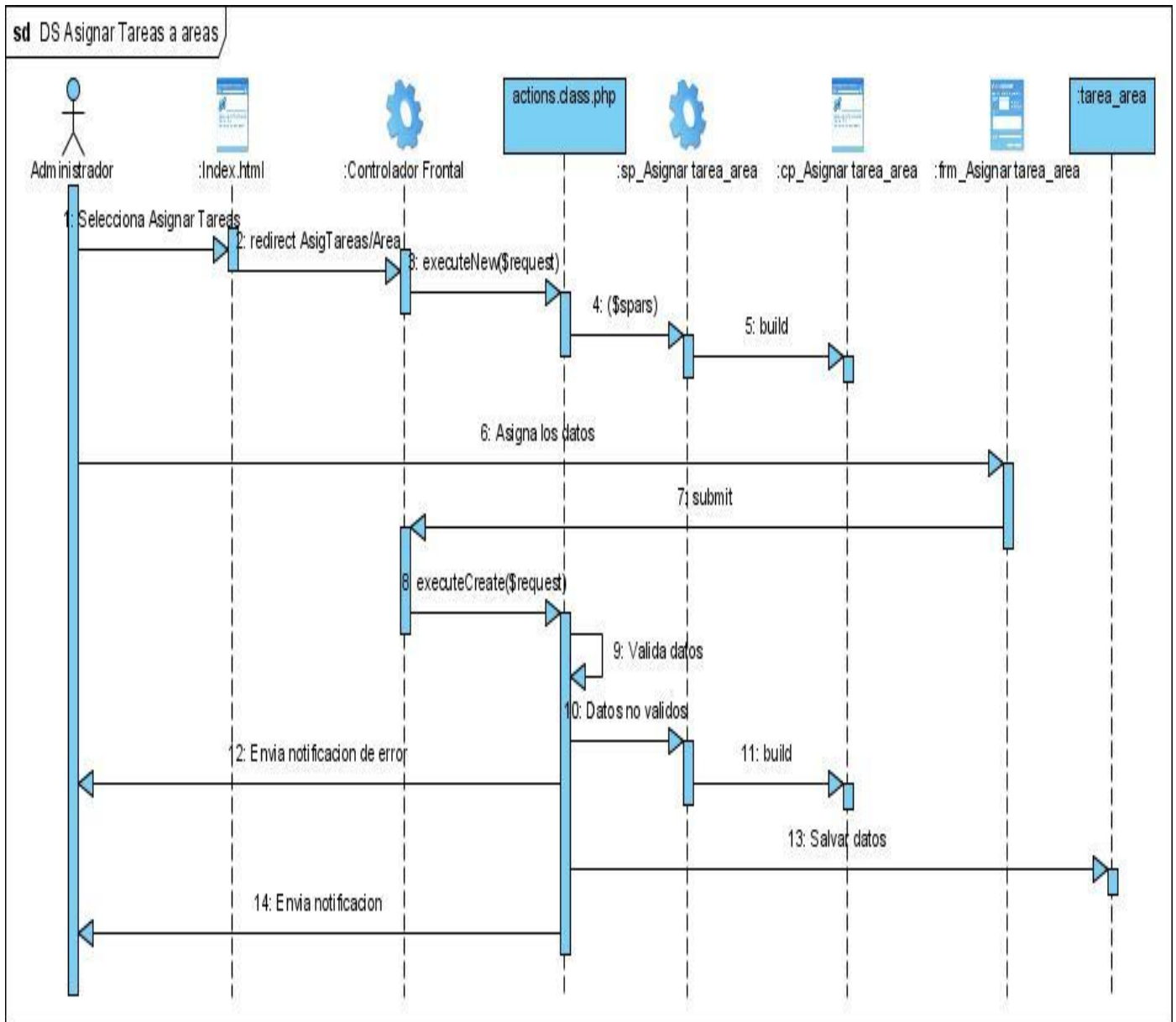


Diagrama de secuencia CU Asignar tarea. Escenario “Asignar tarea a área”.



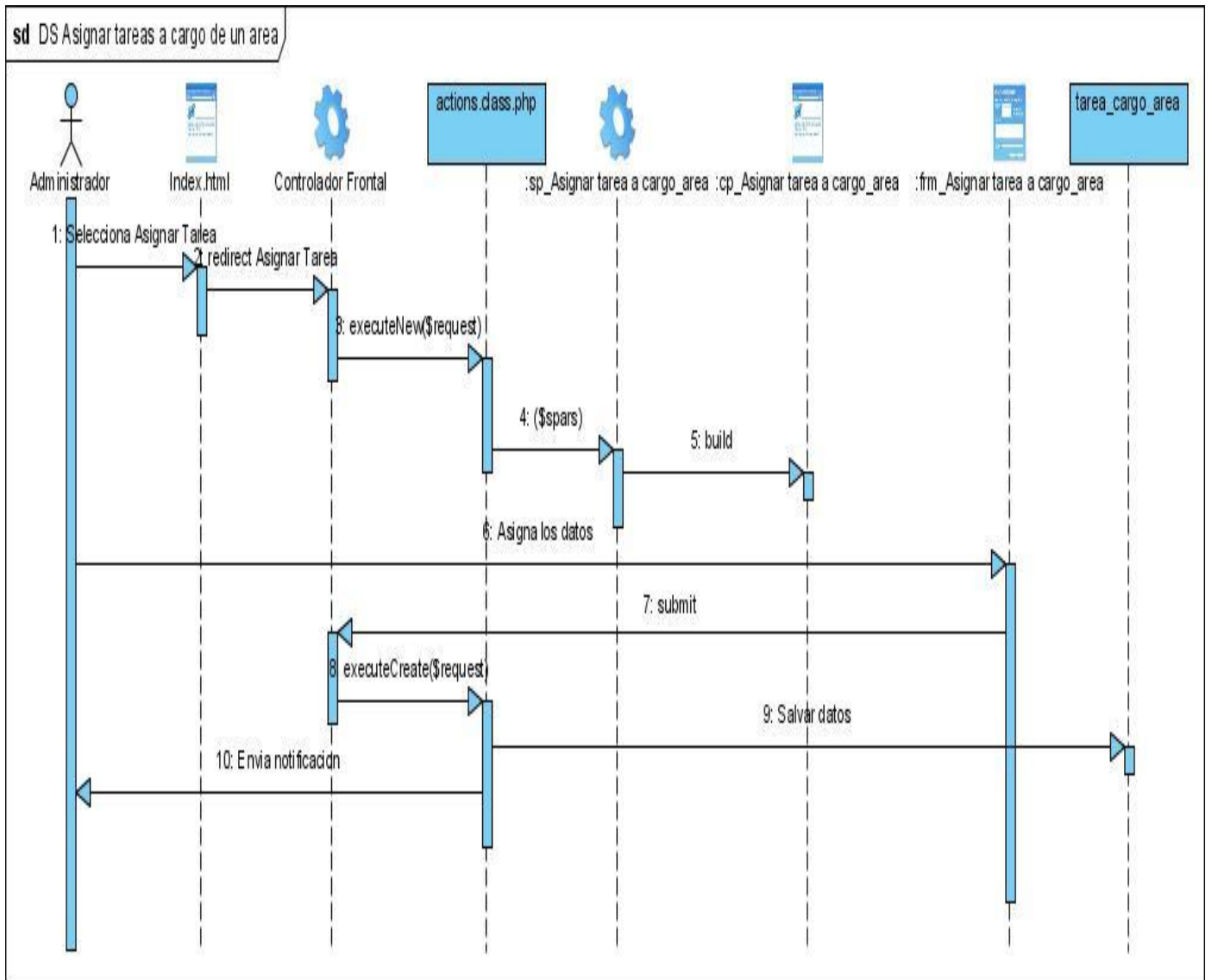


Diagrama de secuencia CU Asignar tarea. Escenario “Asignar tarea a cargo de un área”.

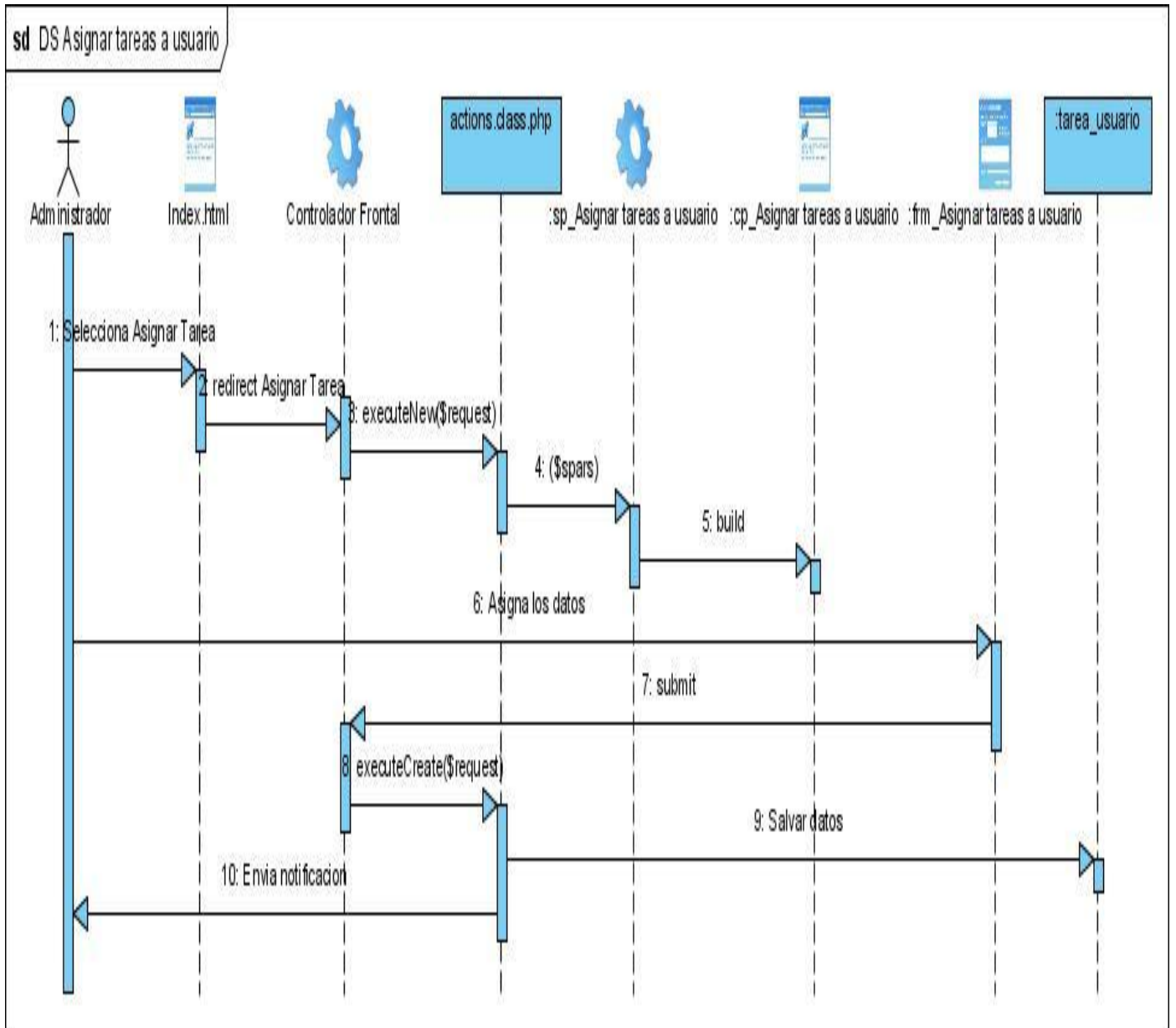


Diagrama de secuencia CU Asignar tarea. Escenario “Asignar tarea a usuario”.

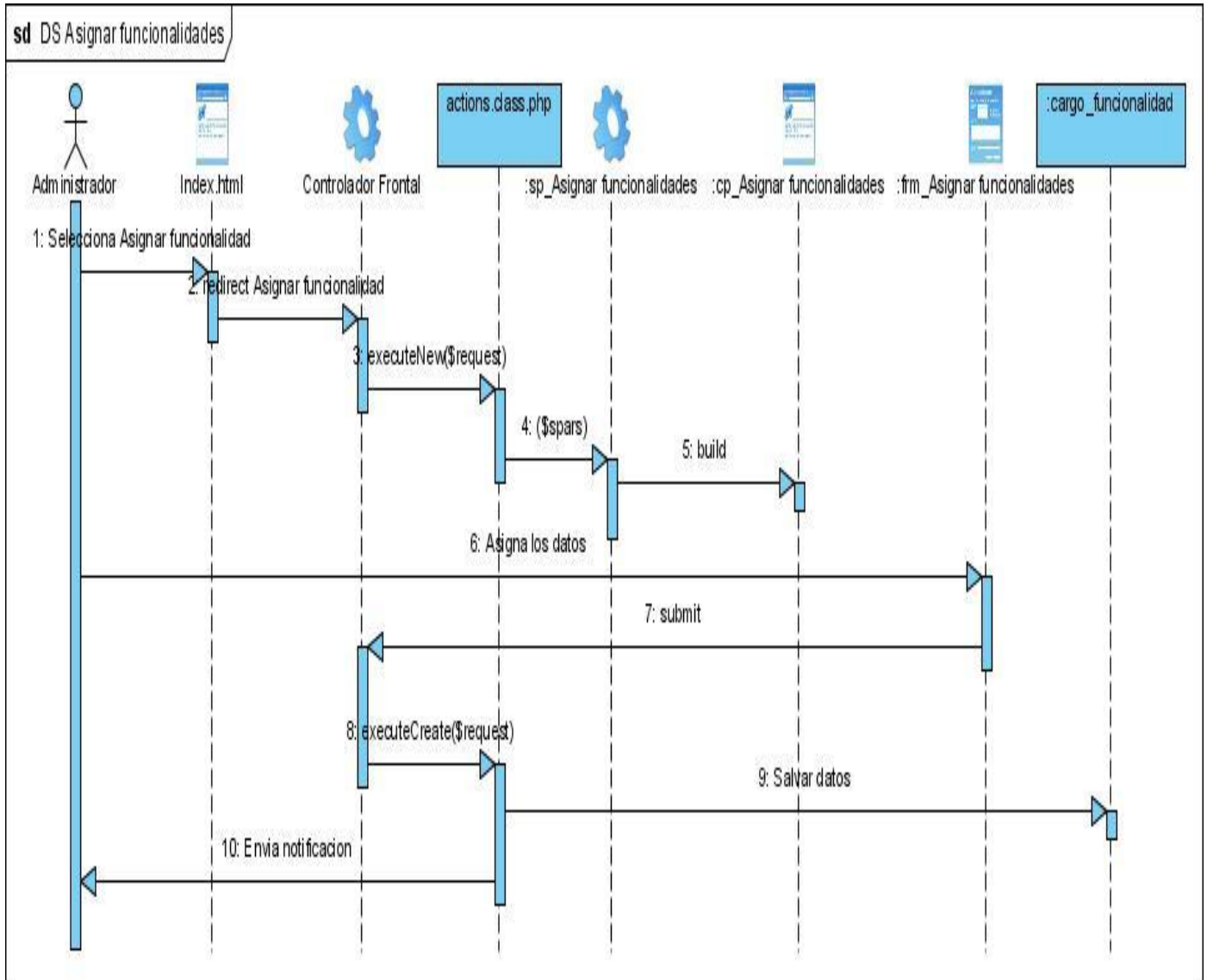


Diagrama de secuencia CU Asignar funcionalidades.

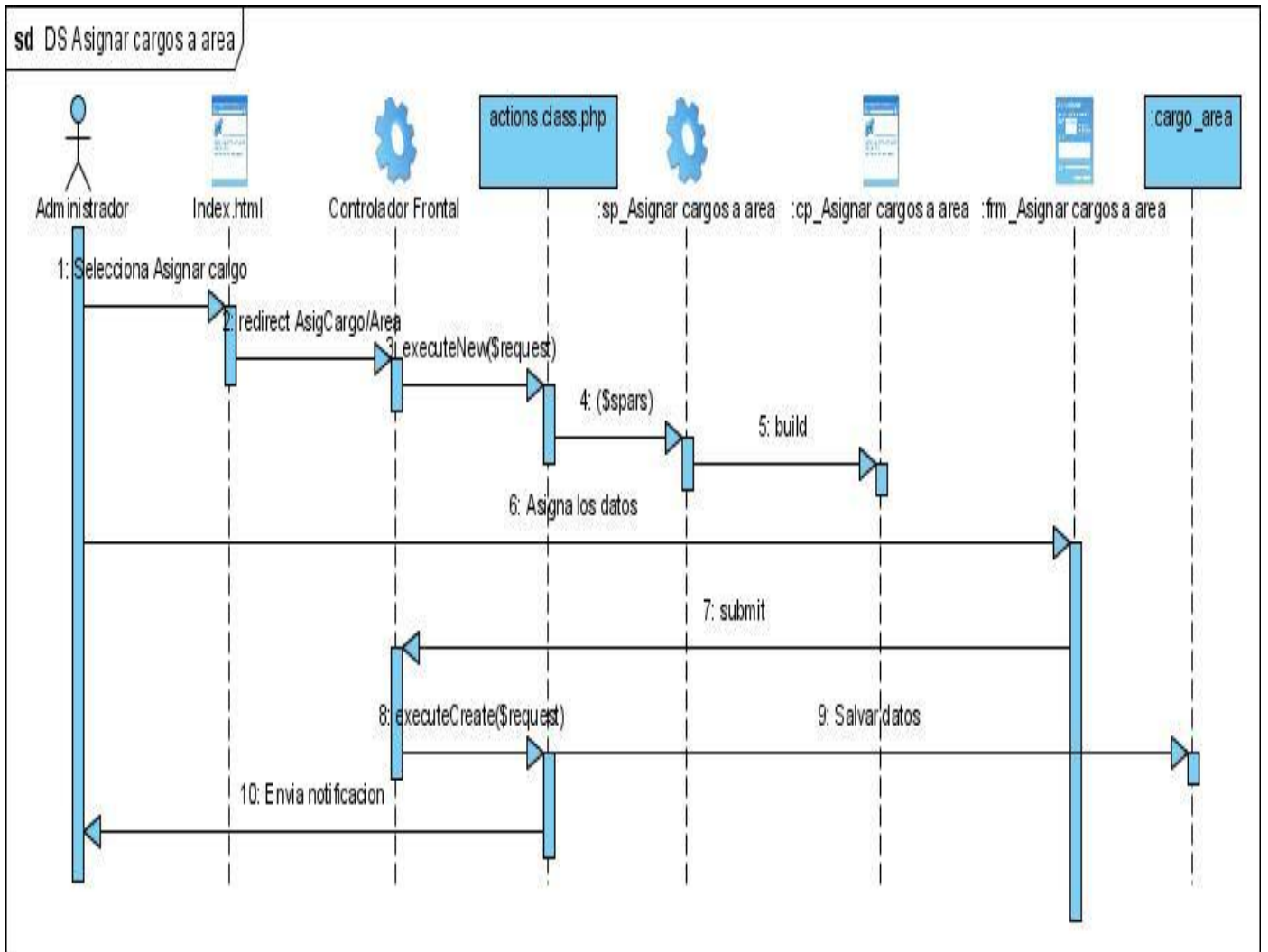


Diagrama de secuencia CU Asignar cargo. Escenario “Asignar cargo a área”.

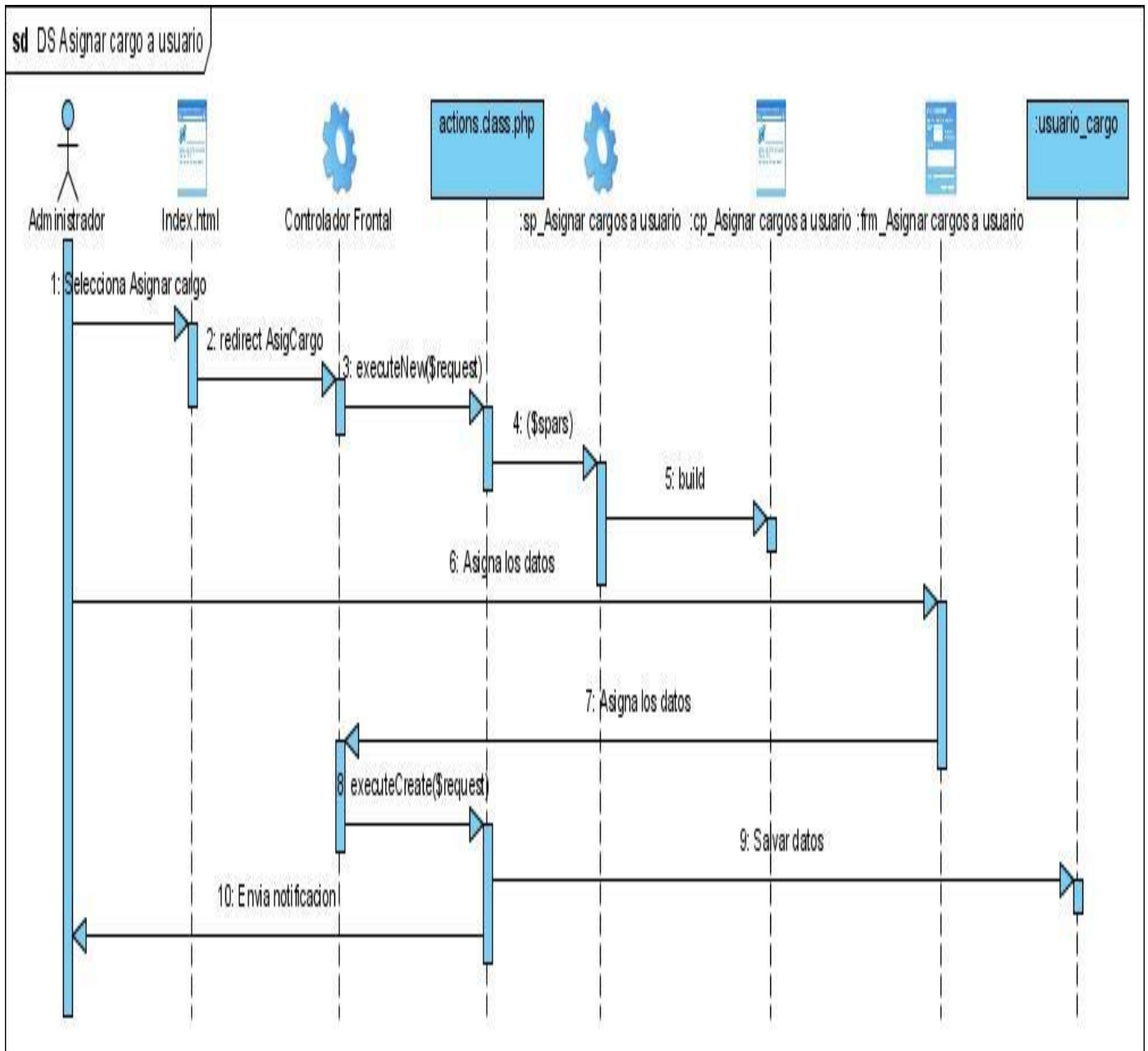


Diagrama de secuencia CU Asignar cargo. Escenario “Asignar cargo a usuario”.

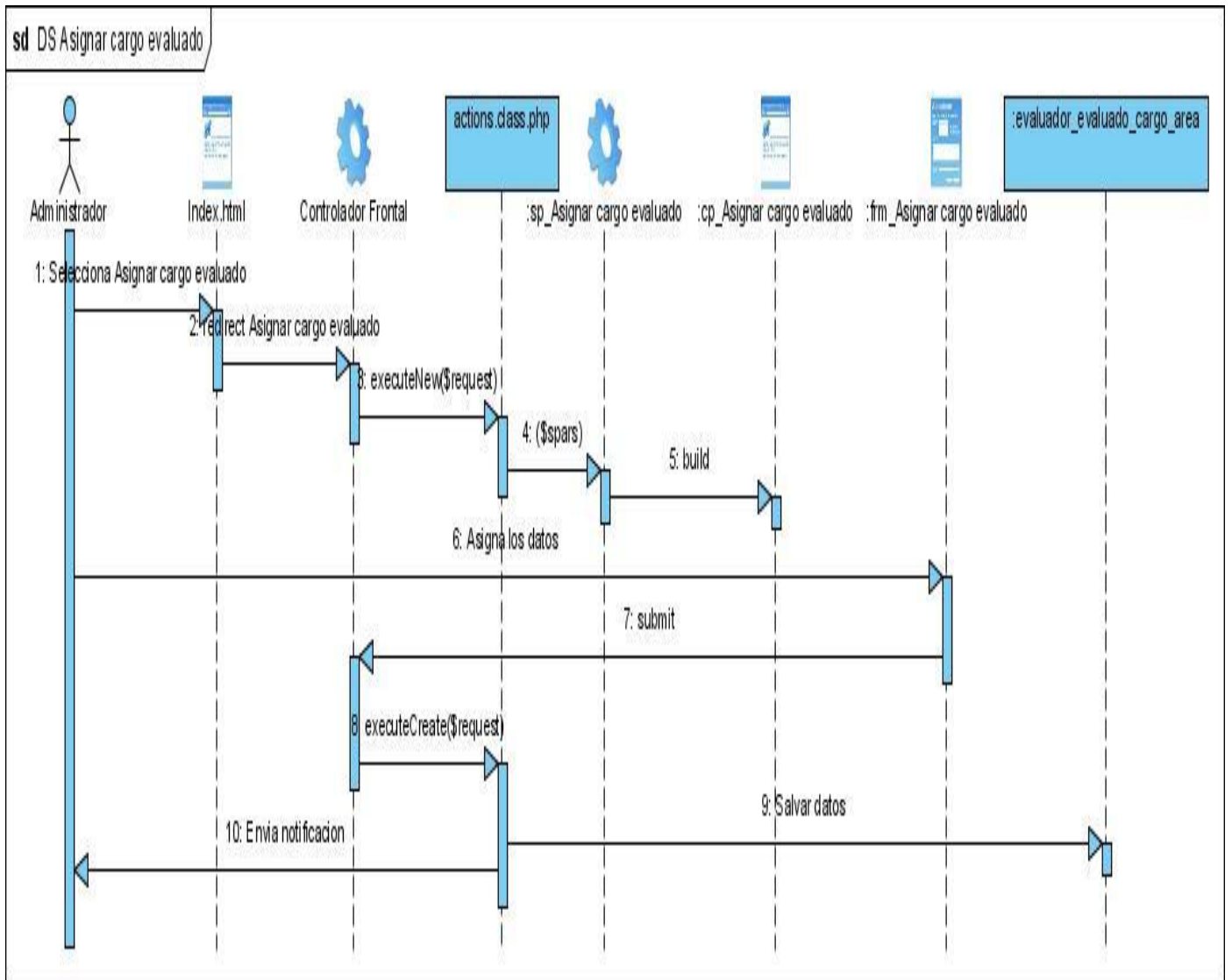


Diagrama de secuencia CU Asignar cargo evaluado.

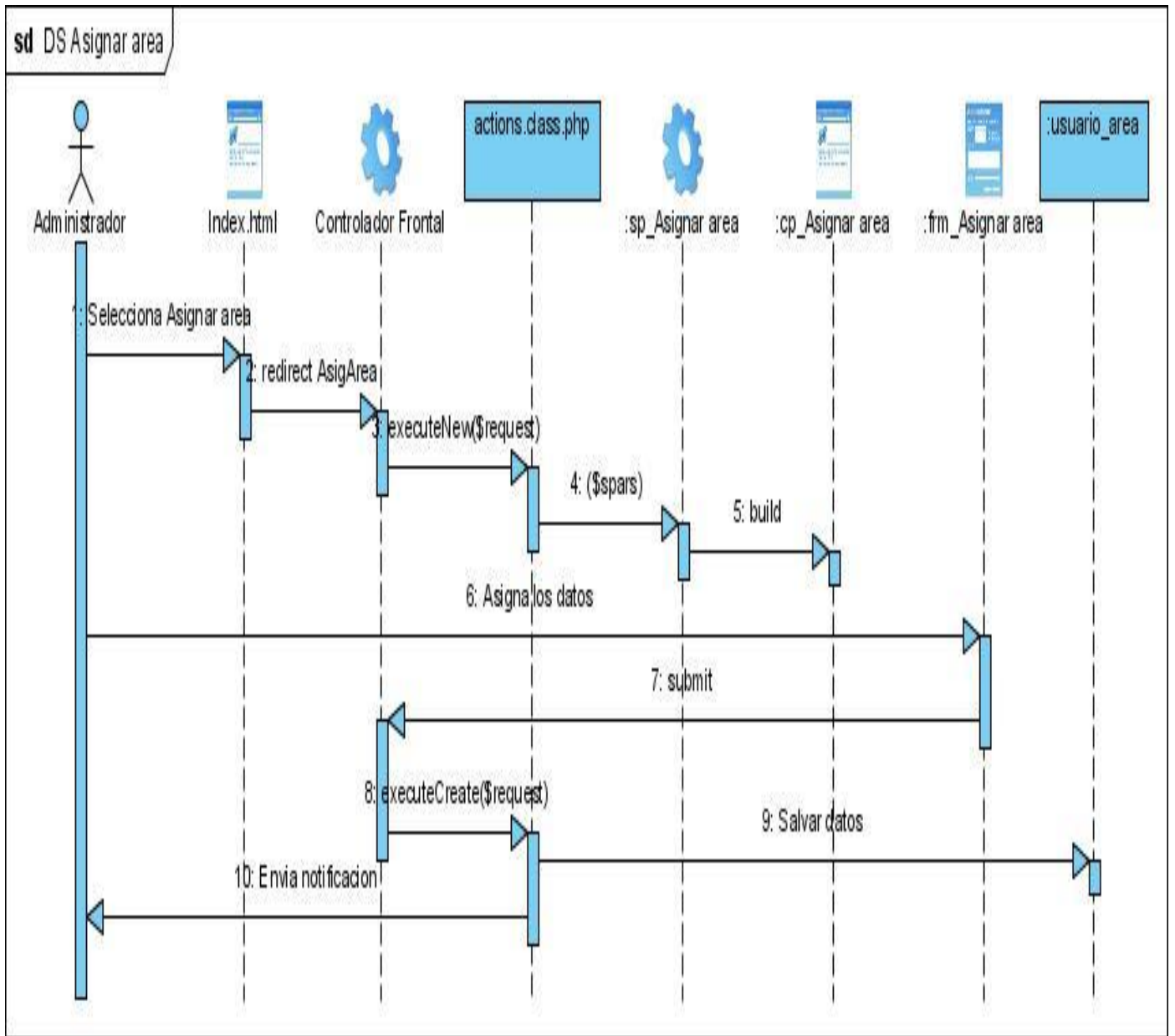


Diagrama de secuencia CU Asignar área.

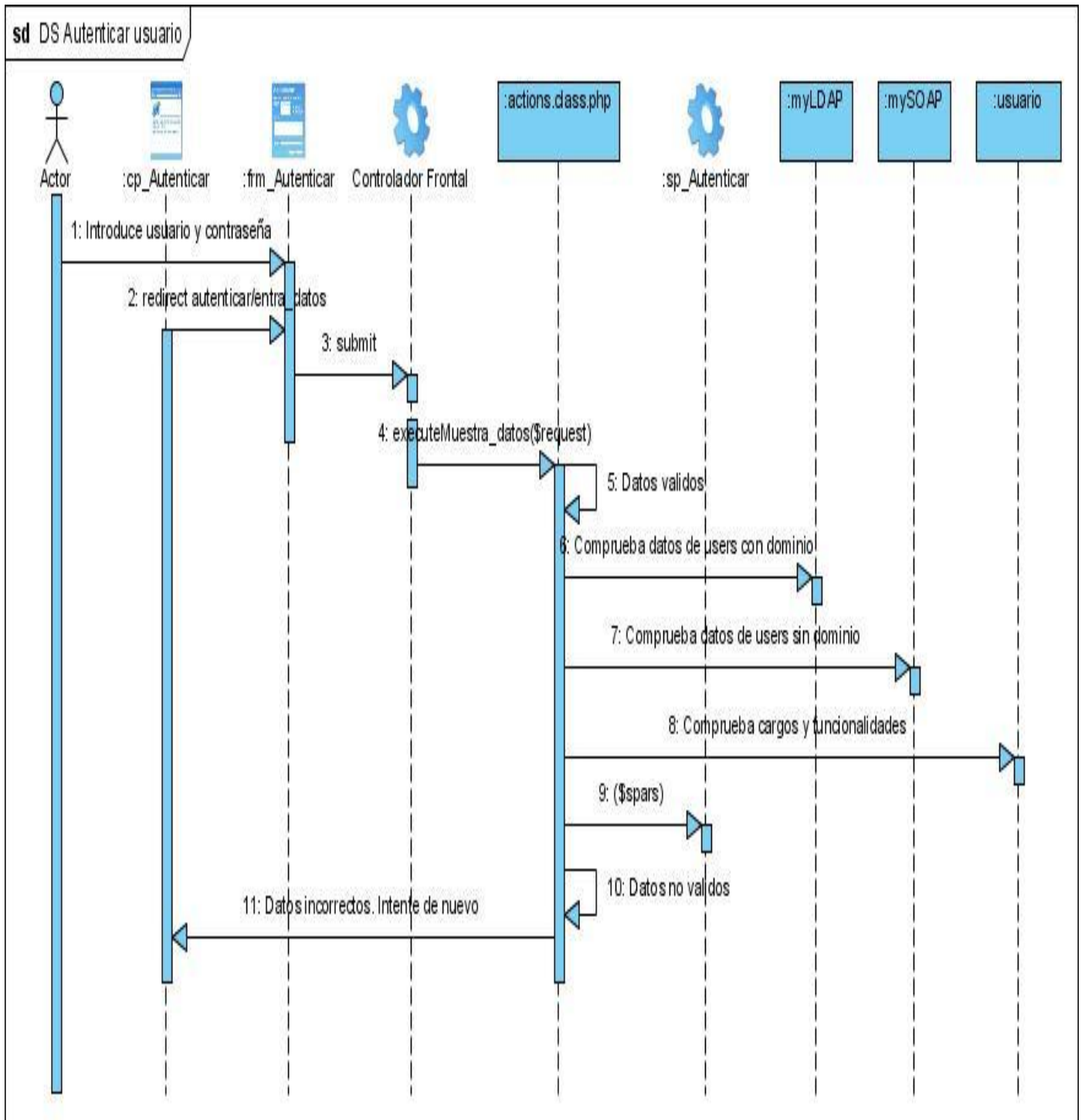


Diagrama de secuencia CU Autenticar usuario.



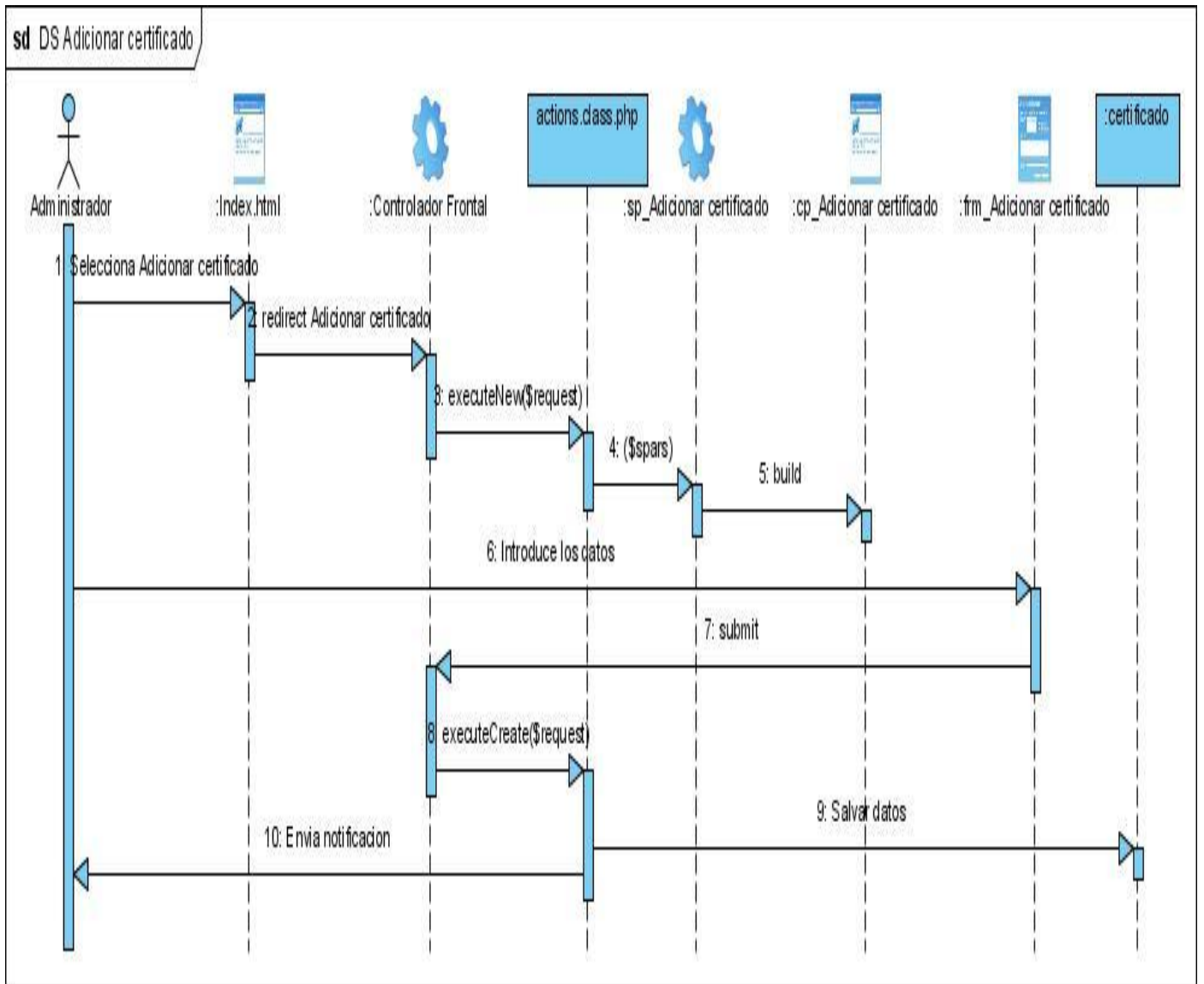


Diagrama de secuencia CU Gestionar certificado. Escenario “Adicionar certificado”.

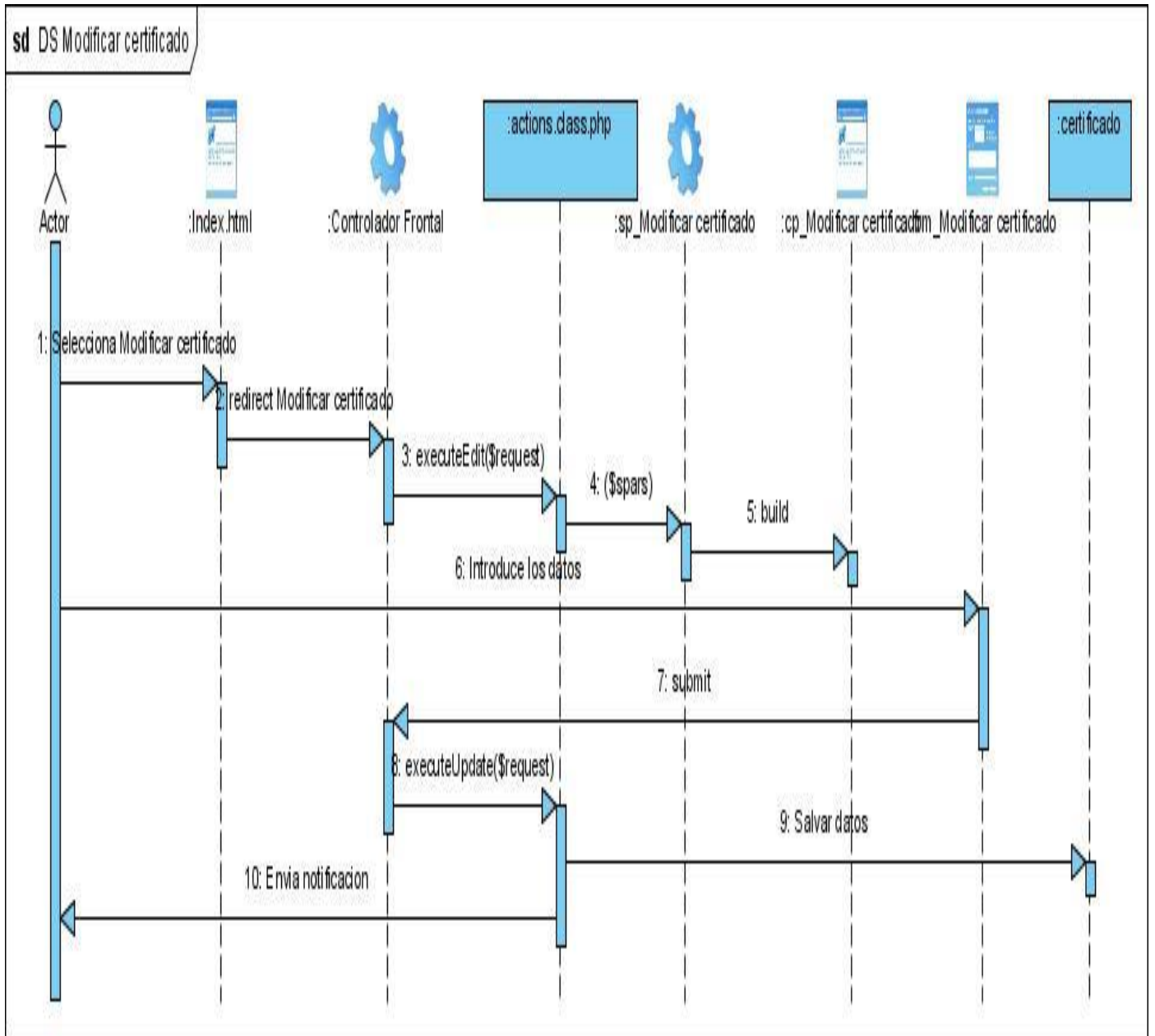


Diagrama de secuencia CU Gestionar certificado. Escenario “Modificar certificado”.

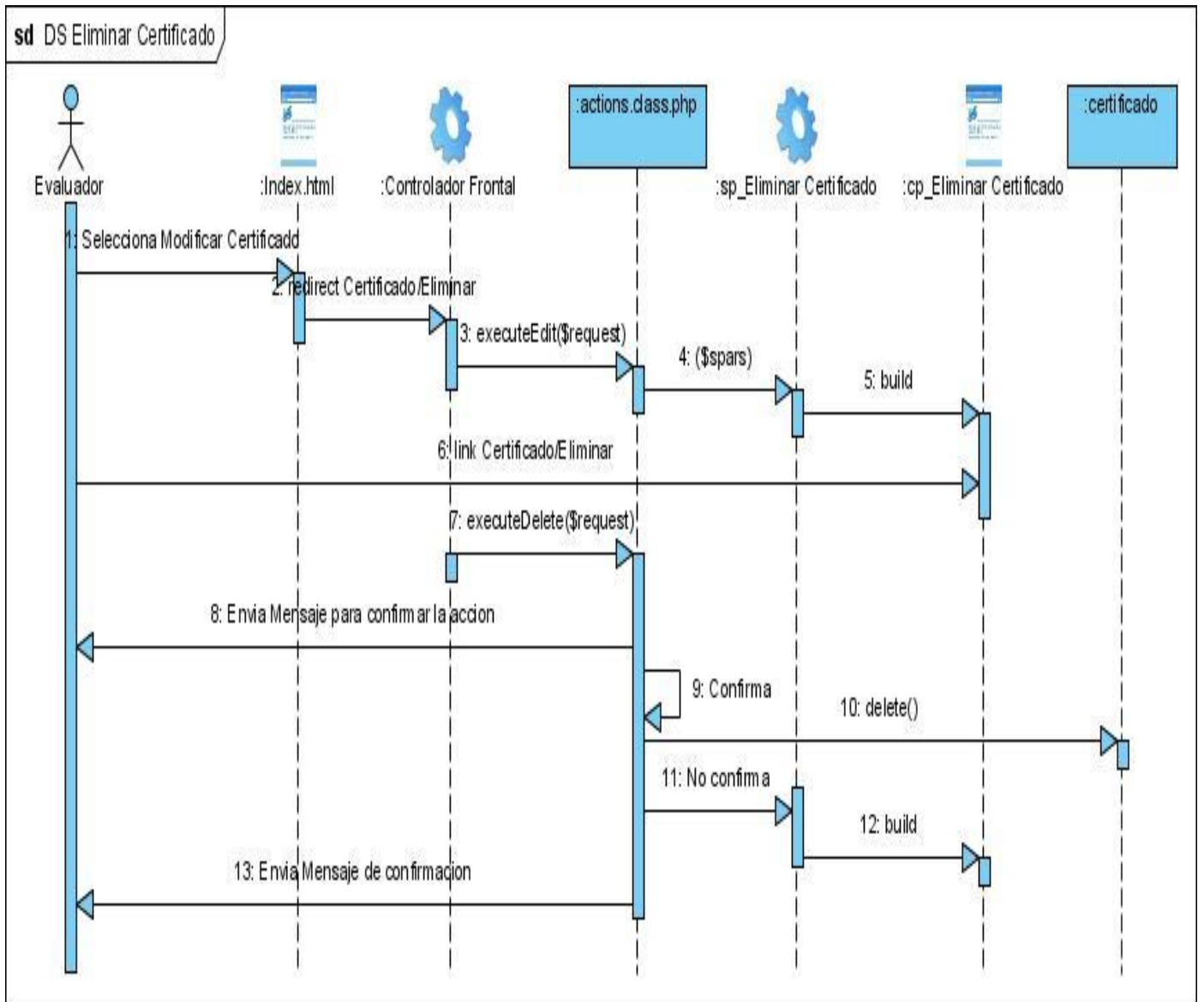


Diagrama de secuencia CU Gestionar certificado. Escenario “Eliminar certificado”.

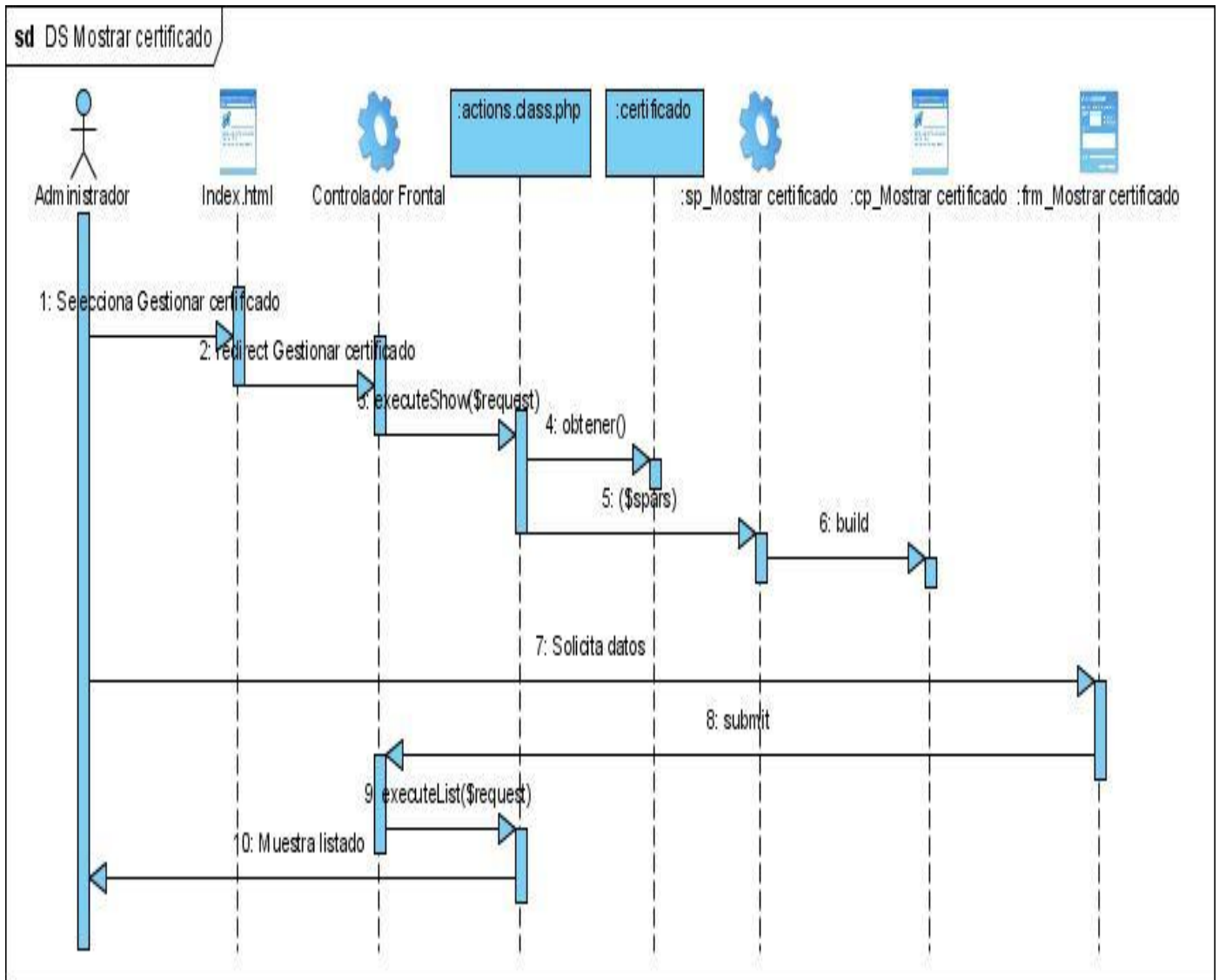


Diagrama de secuencia CU Gestionar certificado. Escenario “Mostrar certificado”.

### Anexo 7: Descripción de los casos de prueba.

#### Caso de uso: Gestionar Tipo de tarea

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado en el sistema con el cargo de webmaster.
El usuario deja el campo Tipo de tarea vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir tipo de tarea".	

Descripción del caso de prueba Gestionar tipo de tarea. Sección "Adicionar tipo de tarea".

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado en el sistema con el cargo de webmaster.
El usuario deja el campo Tipo de tarea vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir tipo de tarea" y se muestra nuevamente la pantalla para modificar.	

Descripción del caso de prueba Gestionar tipo de tarea. Sección "Modificar tipo de tarea".

#### Caso de uso: Gestionar parámetro

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado en el sistema con el cargo de webmaster.
El usuario deja el campo Parámetro vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir parámetro".	

Descripción del caso de prueba Gestionar parámetro. Sección "Adicionar parámetro".

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado en el sistema con el cargo de webmaster.
El usuario deja el campo Parámetro vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir parámetro" y se muestra nuevamente el parámetro a modificar.	

Descripción del caso de prueba Gestionar parámetro. Sección "Modificar parámetro".

**Caso de uso:** Gestionar Tipo de evaluación

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado en el sistema con el cargo de webmaster.
El usuario deja el campo Tipo de tarea vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir tipo de evaluación".	

Descripción del caso de prueba Gestionar tipo de evaluación. Sección "Adicionar tipo de evaluación".

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado en el sistema con el cargo de webmaster.
El usuario deja el campo Tipo de evaluación vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir tipo de evaluación" y se muestra nuevamente la pantalla para modificar.	

Descripción del caso de prueba Gestionar tipo de evaluación. Sección "Modificar tipo de evaluación".

**Caso de uso:** Autenticar usuario

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Entrar</b> .	Se verifican los datos y el usuario tiene acceso a las funcionalidades según su rol.	El usuario debe estar registrado en la base de datos.
El usuario introduce datos incorrectos (usuario o contraseña).	Se muestra el mensaje "Datos incorrectos. Intente de nuevo".	

Descripción del caso de prueba Autenticar usuario.

**Caso de uso:** Gestionar certificado.

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado en el sistema con el cargo de webmaster.
El usuario deja el campo Tipo de tarea vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir tipo de evaluación".	

Descripción del caso de prueba Gestionar certificado. Sección "Adicionar certificado".

Entrada	Resultados	Condiciones
El usuario introduce correctamente los datos y presiona el botón <b>Guardar</b> .	Se guardan los datos y se muestra un mensaje de notificación. "Cambios guardados"	El usuario debe estar autenticado en el sistema con el cargo de webmaster.
El usuario deja el campo Tipo de evaluación vacío y presiona el botón <b>Guardar</b> .	Se muestra el mensaje "Debe escribir tipo de evaluación" y se muestra nuevamente la pantalla para modificar.	

Descripción del caso de prueba Gestionar certificado. Sección "Modificar certificado".